Artificial Intelligence
EDAP01

Lecture 8.2: Natural Language for Communication

Pierre Nugues

Lund University
Pierre.Nugues@cs.lth.se
http://cs.lth.se/pierre_nugues/

February 12, 2021

Syntax

Grammar is the focus of natural language processing in the textbook (Russell and Norvig 2010, Chapter 23).

Two main (modern) traditions: constituent grammars (Chomsky, main advocate) and dependency grammars (Tesnière).

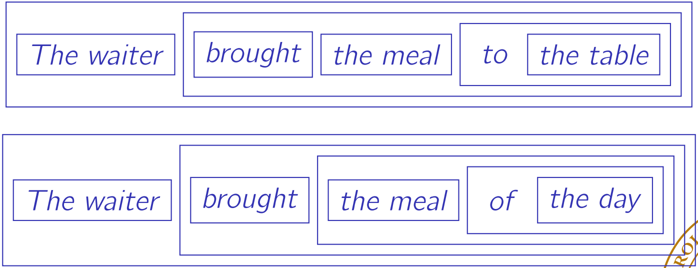
Constituent grammars are still dominant for English, although declining.

But they do not work well for Swedish, as well as many other languages.

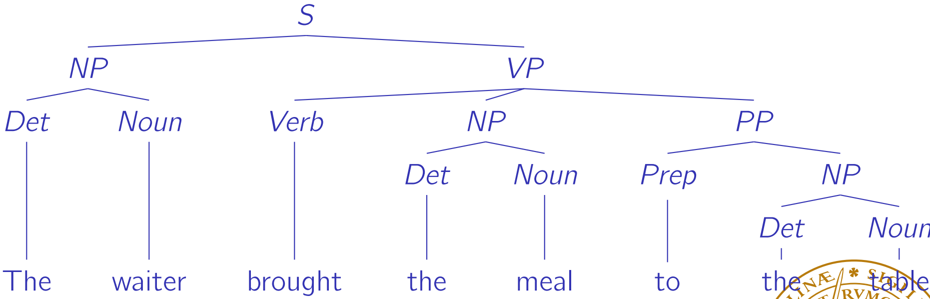
Dependency grammars are more or less universal

Constituents

The waiter brought the meal
The waiter brought the meal to the table
The waiter brought the meal of the day



Syntactic Trees



An Example of PCFG

Rules			P	Rules			P
s	-->	np vp	0.8	det	-->	the	1.0
s	-->	vp	0.2	noun	-->	waiter	0.4
np	-->	det noun	0.3	noun	-->	meal	0.3
np	-->	det adj noun	0.2	noun	-->	day	0.3
np	-->	pronoun	0.3	verb	-->	bring	0.4
np	-->	np pp	0.2	verb	-->	slept	0.2
vp	-->	v np	0.6	verb	-->	brought	0.4
vp	-->	v np pp	0.1	pronoun	-->	he	1.0
vp	-->	v pp	0.2	prep	-->	of	0.6
vp	-->	v	0.1	prep	-->	to	0.4
pp	-->	prep np	1.0	adj	-->	big	1.0

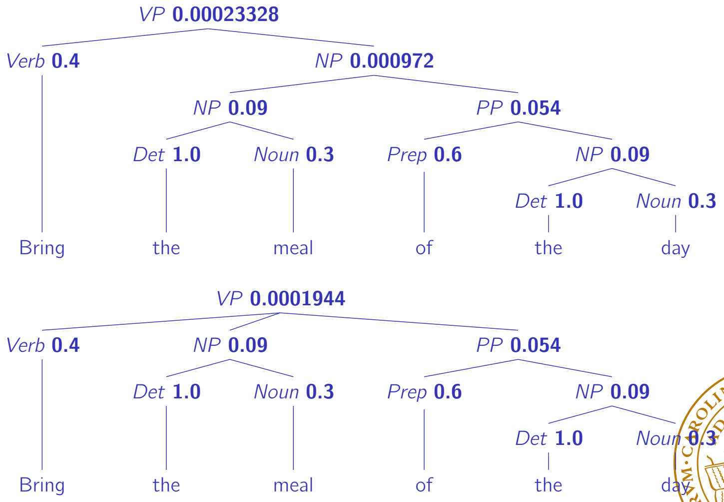
Probabilistic Context-Free Grammars

$$P(T,S) = \prod_{rule(i) \text{producing } T} P(rule(i)).$$

where

$$P(lhs \to rhs_i | lhs) = \frac{Count(lhs \to rhs_i)}{\sum_j Count(lhs \to rhs_j)}.$$

Computing the Probabilities



Semantic Parsing

Converts sentences to first-order logic or predicate-argument structures

Example:

Mr. Schmidt called Bill

to

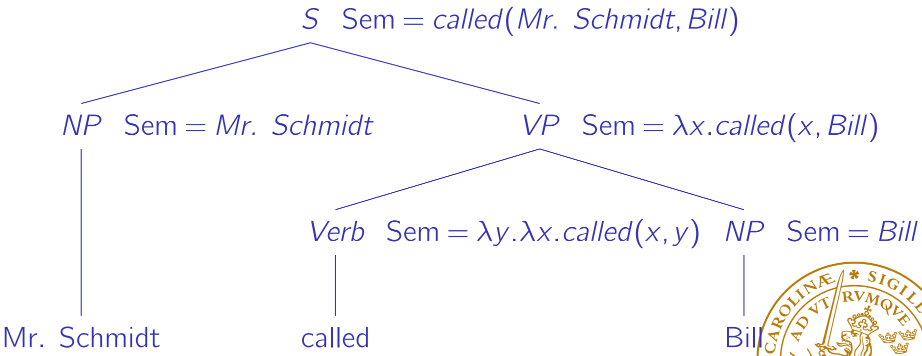
```
called('Mr. Schmidt', 'Bill').
```

Assumption: We can compose sentence fragments (phrases) into logical forms while parsing

This corresponds to the compositionality principle

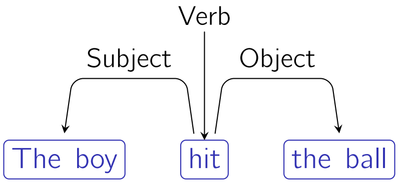
Semantic Composition

Semantic composition can be viewed as a parse tree annotation

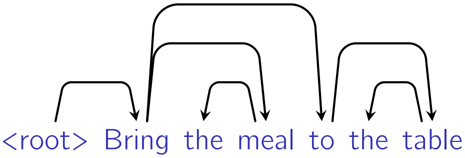


The Current Approach: Dependencies

A graph of dependencies and functions:

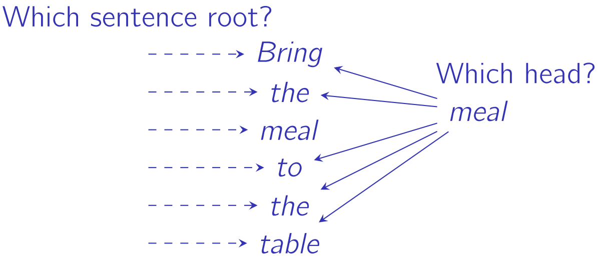


Conventions: Each word has a head and the main word is linked to an artificial root:



Parsing Dependencies

Generate all the pairs:



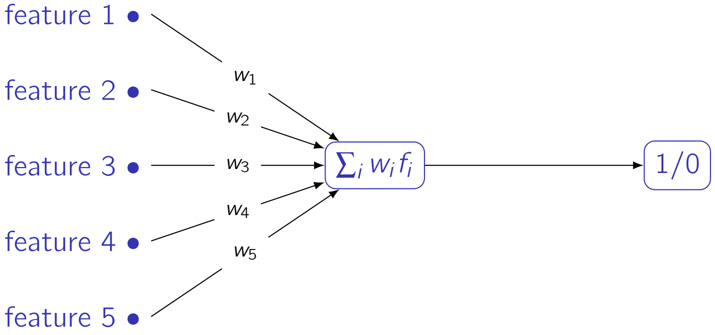
Algorithms: Extensions to shift-reduce or graph optimization trained on annotated corpora.
Corpora: `https://universaldependencies.org/`

Neural Networks: Representation

Many NLP tasks involve classifiers, more and more relying on neural networks.

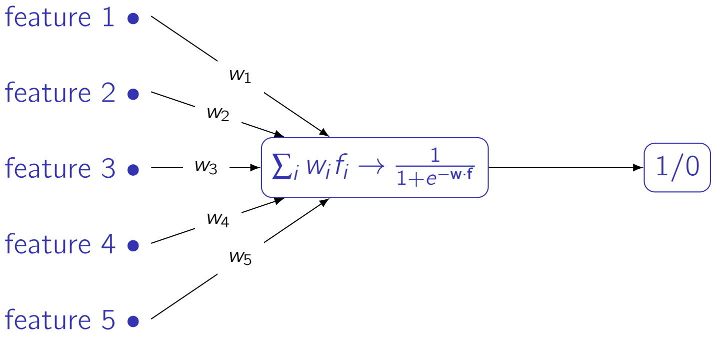
For instance, in POS tagging, is the word table a verb?

The base network: An input layer and an output layer (perceptron):



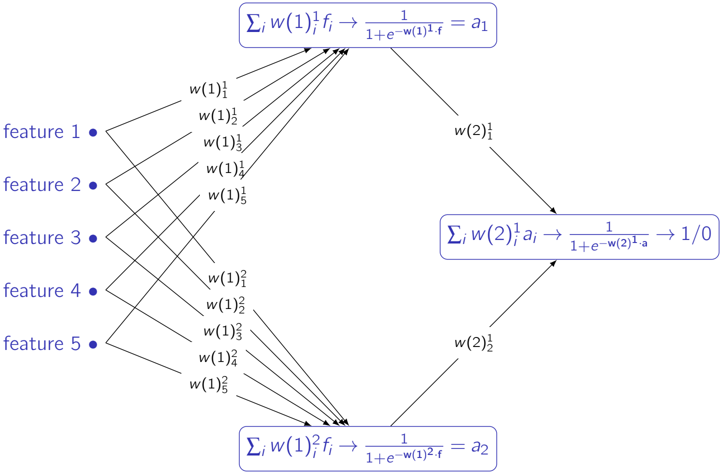
Neural Networks: Activation Function

And logistic regression:



The logistic function is the activation function of the node

Neural Networks: Hidden Layers



Demonstration: `http://playground.tensorflow.org/`

Input: Word Embeddings (I)

In most cases, the input consists of dense numerical vectors: the embeddings

Many way to create such embeddings: We describe here GloVe

We store the word-context pairs (w_i, C_j) in a matrix, where C_j is a small window of words

Mutual information, often called pointwise mutual information:

$$I(w_i, w_j) = \log_2 \frac{P(w_i, w_j)}{P(w_i)P(w_j)} \approx \log_2 \frac{N \cdot C(w_i, w_j)}{C(w_i)C(w_j)}.$$

Words\D#	w_1	w_2	w_3	...	w_n
w_1	$MI(w_1, w_1)$	$MI(w_1, w_2)$	$MI(w_1, w_3)$	...	$MI(w_1, w_n)$
w_2	$MI(w_2, w_1)$	$MI(w_2, w_2)$	$MI(w_2, w_3)$	...	$MI(w_2, w_n)$
w_3	$MI(w_3, w_1)$	$MI(w_3, w_2)$	$MI(w_3, w_3)$	...	$MI(w_3, w_n)$
...	...	...	...	...	
w_m	$MI(w_m, w_1)$	$MI(w_m, w_2)$	$MI(w_m, w_3)$	...	$MI(w_m, w_n)$

Input: Word Embeddings (II)

Words\D#	w_1	w_2	w_3	...	w_n
w_1	$MI(w_1, w_1)$	$MI(w_1, w_2)$	$MI(w_1, w_3)$	...	$MI(w_1, w_n)$
w_2	$MI(w_2, w_1)$	$MI(w_2, w_2)$	$MI(w_2, w_3)$	...	$MI(w_2, w_n)$
w_3	$MI(w_3, w_1)$	$MI(w_3, w_2)$	$MI(w_3, w_3)$	...	$MI(w_3, w_n)$
...	...	...	...	...	...
w_m	$MI(w_m, w_1)$	$MI(w_m, w_2)$	$MI(w_m, w_3)$	...	$MI(w_m, w_n)$

We apply a principal component analysis to reduce the dimensions to 50, 100, or 300.

Machine Translation

Natural language processing was born with machine translation

Massive advance when the US government decided to fund large-scale translation programs to have a quick access to documents written in Russian

IBM teams pioneered statistical models for machine translation in the early 1990s

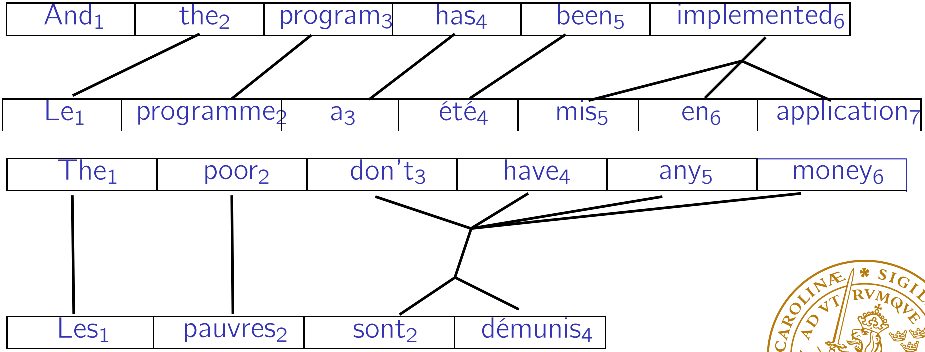
Their work that used the English (e) and French (f) parallel versions of the Canadian Hansards is still the standard reference in the field.

Parallel Corpora (Swiss Federal Law)

German	French	Italian
Art. 35 Milchtransport	**Art. 35 Transport du lait**	**Art. 35 Trasporto del latte**
1 Die Milch ist schonend und hygienisch in den Verarbeitungsbetrieb zu transportieren. Das Transportfahrzeug ist stets sauber zu halten. Zusammen mit der Milch dürfen keine Tiere und milchfremde Gegenstände transportiert werden, welche die Qualität der Milch	1 Le lait doit être transporté jusqu'à l'entreprise de transformation avec ménagement et conformément aux normes d'hygiène. Le véhicule de transport doit être toujours propre. Il ne doit transporter avec le lait aucun animal ou objet susceptible d'en altérer la qualité.	1 Il latte va trasportato verso l'azienda di trasformazione in modo accurato e igienico. Il veicolo adibito al trasporto va mantenuto pulito. Con il latte non possono essere trasportati animali e oggetti estranei, che potrebbero pregiudicarne la qualità.

Alignment (Brown et al. 1993)

Canadian Hansard



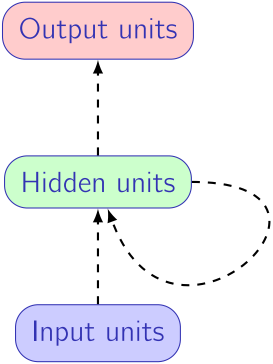
Machine Translation Algorithms

A statistical model:

$$P(f, d|e) = \prod_i P(f_i|e_i)P(d_i),$$

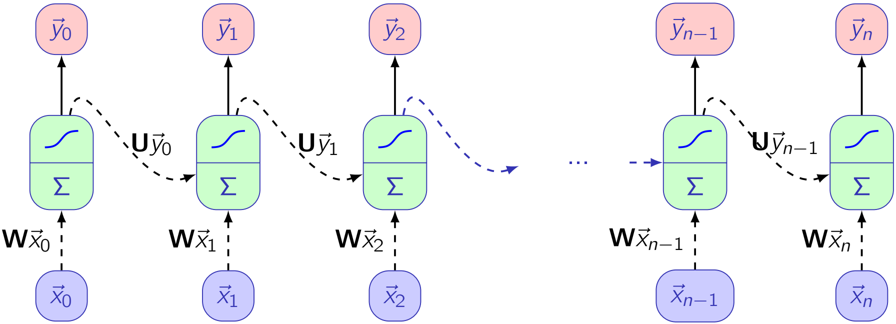
where d measures the distortion, how much reassembling is needed from English to French.
Distortion has the form of a right-to-left or left-to-right shift.
Recently, recurrent neural network architectures improved considerably the performance of machine translation.

Recurrent Neural Networks



A simple recurrent neural network; the dashed lines represent trainable connections.

The Unfolded RNN Architecture



The network unfolded in time. Equation used by implementations[1]:

$$\mathbf{y}_{(t)} = \tanh(\mathbf{W} \cdot \mathbf{x}_{(t)} + \mathbf{U} \cdot \mathbf{y}_{(t-1)} + \mathbf{b})$$

[1] See: https://pytorch.org/docs/stable/nn.html#torch.nn.RNN

LSTMs

Simple RNNs use the previous output as input. They have then a very limited feature context.

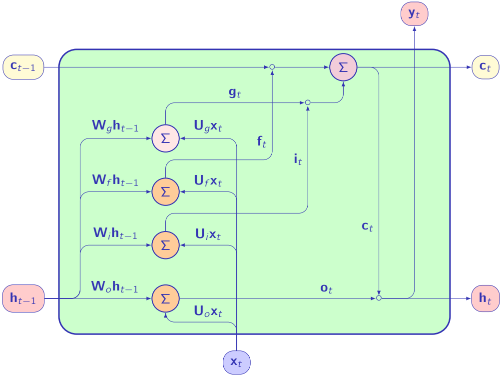
Long short-term memory units (LSTM) are an extension to RNNs that can remember, possibly forget, information from longer or more distant sequences.

Given an input at index t, $\mathbf{x}_t$, a LSTM unit produces:

- A short term state, called $\mathbf{h}_t$ and
- A long-term state, called $\mathbf{c}_t$ or memory cell.

The short-term state, $\mathbf{h}_t$, is the unit output, i.e. $\mathbf{y}_t$; but both the long-term and short-term states are reused as inputs to the next unit.

The LSTM Architecture



An LSTM unit showing the data flow, where $\mathbf{g}_t$ is the unit input, $\mathbf{i}_t$, the input gate, $\mathbf{f}_t$, the forget gate, and $\mathbf{o}_t$, the output gate. The activation functions have been omitted

Speech Recognition

Conditions to take into account:

- Number of speakers
- Fluency of speech.
- Size of vocabulary
- Syntax
- Environment

Structure of Speech Recognition

Words:

$$W = w_1, w_2, \ldots, w_n.$$

Acoustic symbols:

$$A = a_1, a_2, \ldots, a_m,$$

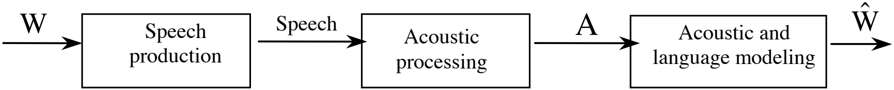
$$\hat{W} = \arg\max_{W} P(W|A).$$

Using Bayes' formula,

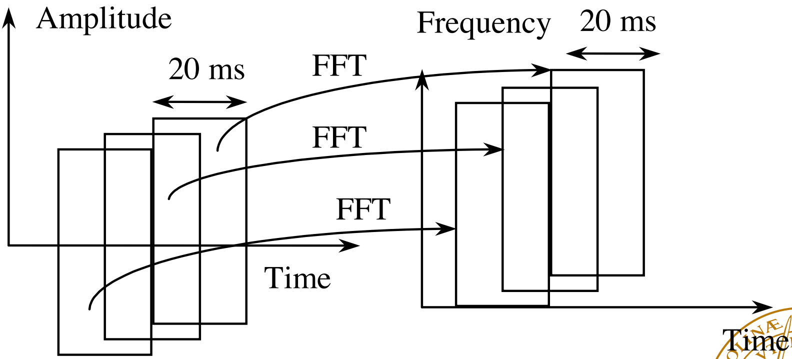
$$P(W|A) = \frac{P(A|W)P(W)}{P(A)}.$$

Two-Step Recognition

$$\hat{W} = \arg\max_{W} P(A|W)P(W).$$

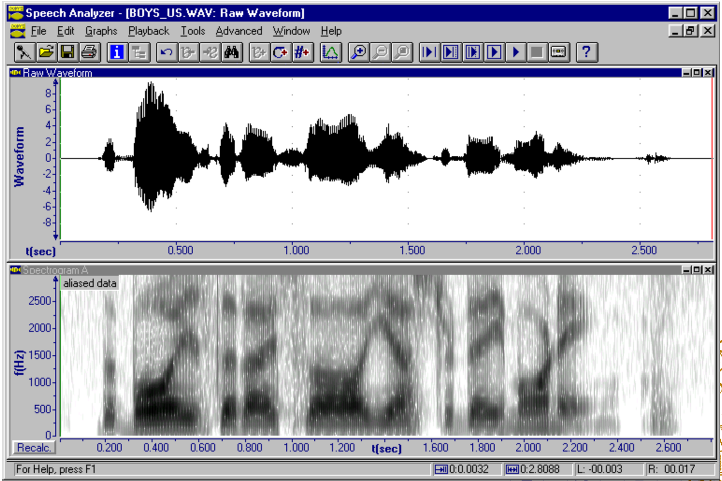


Speech Spectrograms



Speech Signals

The boys I saw yesterday morning



Neural Networks for Speech Recognition (I)

From 2015-2016, neural network architectures started to overtake HMM.
Most current systems use variants of recurrent neural networks.
A historical model from Waibel et al., Phoneme recognition using time-delay neural networks, 1989.

- Three phonemes B, D, and G
- An input vector consists of 16 melscale coefficients from a Fourier transform of a speech window of 10 ms: Energy at certain frequencies
- The context is modeled as a sequence of three such input vectors.
- Two hidden layers

Neural Networks for Speech Recognition (II)

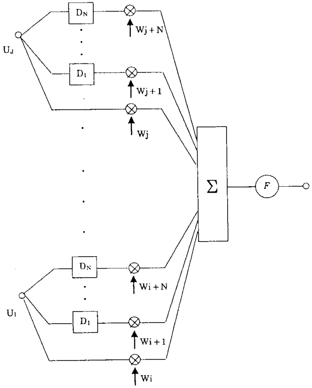


Fig. 1. A Time-Delay Neural Network (TDNN) unit.

From Waibel et al., Phoneme recognition using time-delay neural networks, IEEE Transactions of Acoustics, Speech, and Signal

Neural Networks for Speech Recognition (III)

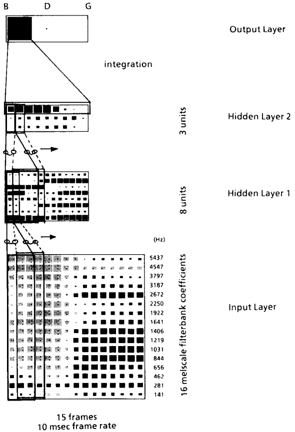


Fig. 2. The architecture of the TDNN.

From Waibel et al., Phoneme recognition using time-delay neural networks, IEEE Transactions of Acoustics, Speech, and Signal