FMAN45
Maskininlärning
http://cs.lth.se/edan20/
Modern Techniques in Natural Language Processing

Pierre Nugues

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

May 23, 2019

Applications of Language Processing

- Spelling and grammatical checkers: *MS Word*, e-mail programs, etc.
- Text indexing and information retrieval on the Internet: *Google*, *Microsoft Bing*, *Yahoo*, or software like *Apache Lucene*
- Translation: *Google Translate*, *Bing Translate*
- Spoken interaction: Apple Siri, Google Now, *Tellme.com*, or *SJ* (trains in Sweden)
- Speech dictation of letters or reports: *IBM ViaVoice*, *Windows*

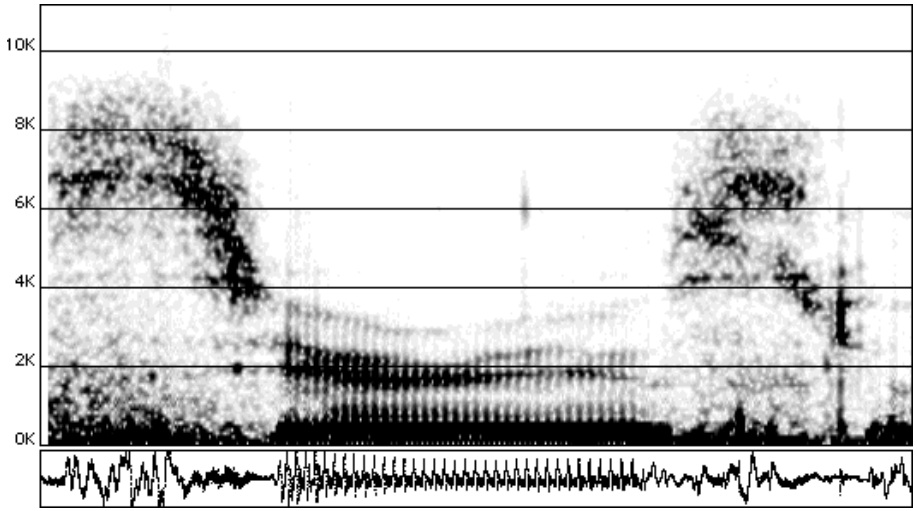
Applications of Language Processing (ctn'd)

- Direct translation from spoken English to spoken Swedish in a restricted domain: *SRI* and *SICS*
- Voice control of domestic devices such as tape recorders: *Philips* or disc changers: *MS Persona*
- Conversational agents able to dialogue and to plan: *TRAINS*
- Spoken navigation in virtual worlds: *Ulysse*, *Higgins*
- Generation of 3D scenes from text: *Carsim*
- Question answering: *IBM Watson* and *Jeopardy!*

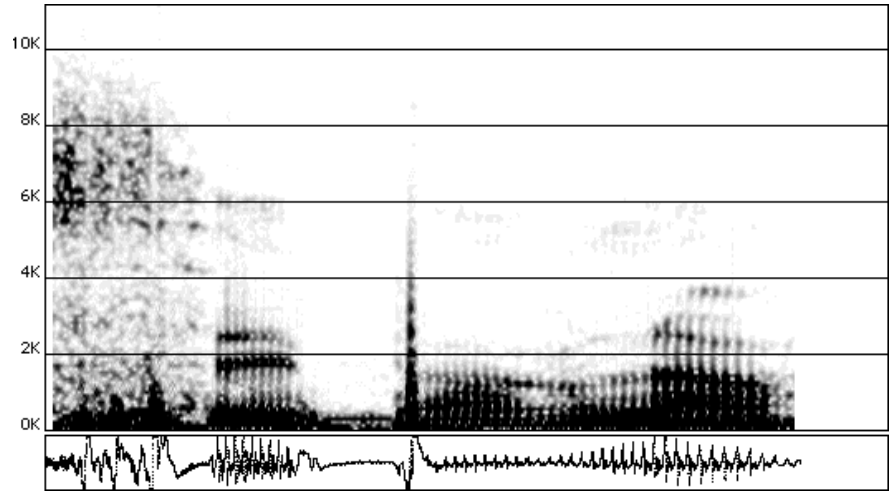
Linguistics Layers

- Sounds
- Phonemes
- Words and morphology
- Syntax and functions
- Semantics
- Dialogue

Sounds and Phonemes



Serious



C'est par là 'It is that way'

Lexicon and Parts of Speech

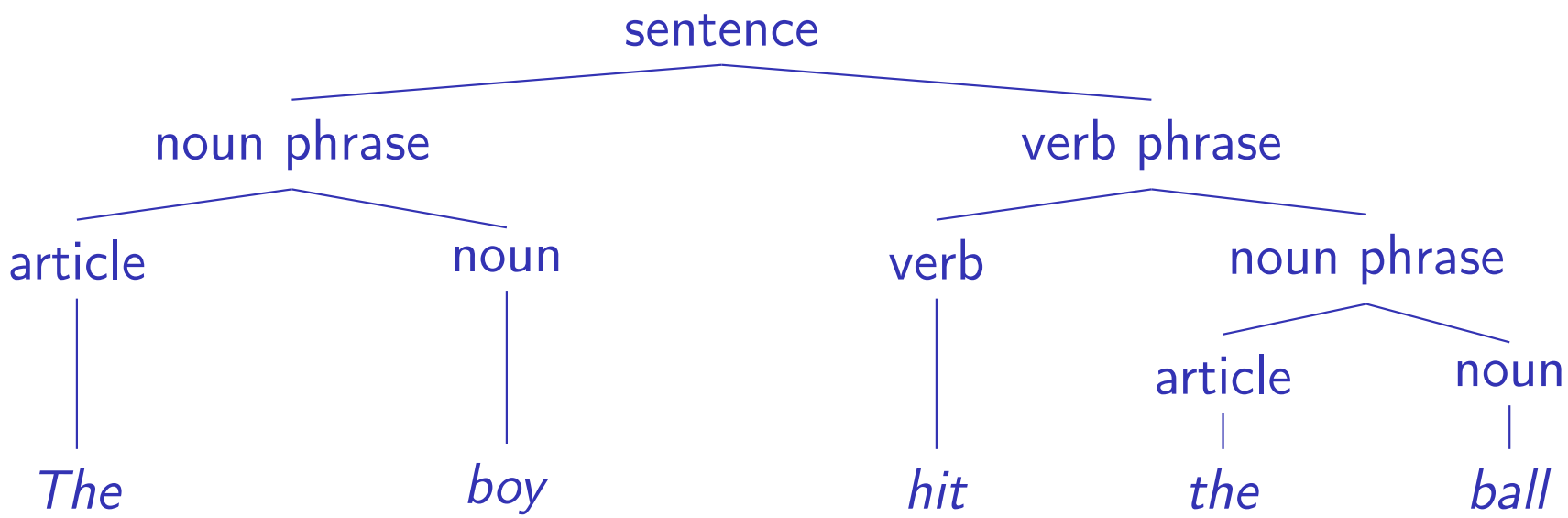
The big cat ate the gray mouse

The/article *big*/adjective *cat*/noun *ate*/verb *the*/article *gray*/adjective *mouse*/noun
Le/article *gros*/adjectif *chat*/nom *mange*/verbe *la*/article *souris*/nom *grise*/adjectif
Die/Artikel *große*/Adjektiv *Katze*/Substantiv *ißt*/Verb *die*/Artikel *graue*/Adjektiv *Maus*/Substantiv

Morphology

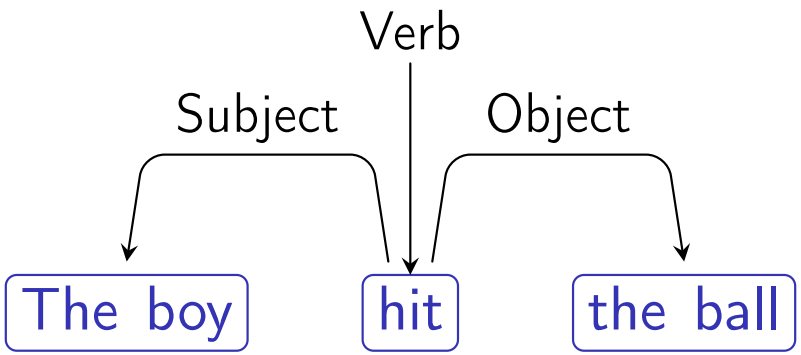
Word	Root form
worked	*to work* + verb + preterit
travaillé	*travailler* + verb + past participle
gearbeitet	*arbeiten* + verb + past participle

Syntactic Tree



Syntax: A Classical View

A graph of dependencies and functions



Semantics

As opposed to syntax:

1. Colorless green ideas sleep furiously.
2. *Furiously sleep ideas green colorless.

Determining the logical form:

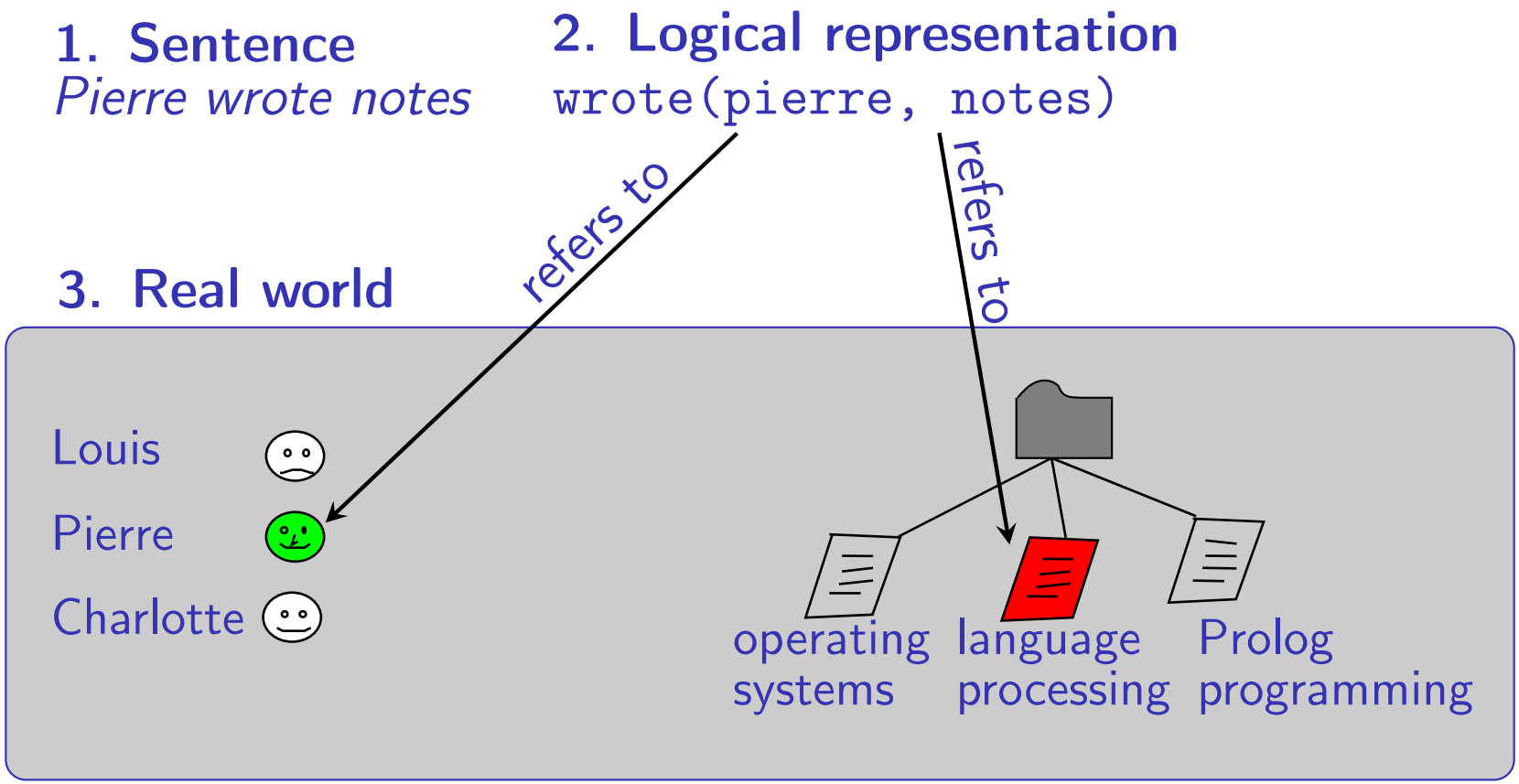
Sentence	Logical representation
Frank is writing notes	`writing(Frank, notes).`
François écrit des notes	`écrit(François, notes).`
Franz schreibt Notizen	`schreibt(Franz, Notizen).`

Lexical Semantics

Word senses:

1. **note** (*noun*) short piece of writing;
2. **note** (*noun*) a single sound at a particular level;
3. **note** (*noun*) a piece of paper money;
4. **note** (*verb*) to take notice of;
5. **note** (*noun*) of note: of importance.

Reference


1. Sentence
Pierre wrote notes
2. Logical representation
wrote(pierre, notes)
refers to
refers to
3. Real world
Louis
Pierre
Charlotte
operating systems
language processing
Prolog programming

Ambiguity

Many analyses are ambiguous. It makes language processing difficult.
Ambiguity occurs in any layer: speech recognition, part-of-speech tagging, parsing, etc.
Example of an ambiguous phonetic transcription:
The boys eat the sandwiches
That may correspond to:
The boy seat the sandwiches; *the boy seat this and which is*; *the buoys eat the sand which is*

Models and Tools

Linguistics has produced an impressive set of theories and models
Language processing requires significant resources
Models and tools have matured. Resources are available.
Tools involve notably finite-state automata, regular expressions, logic, statistics, and machine learning.
Demo: Langforia `http://vilde.cs.lth.se:9000/#/`

The Carsim System: A Text-to-Scene Converter

Texts	XML Templates	3D Animation

Véhicule B venant de ma gauche, je me trouve dans le carrefour, à faible vitesse environ 40 km/h, quand le véhicule B, percute mon véhicule, et me refuse la priorité à droite. Le premier choc atteint mon aile arrière gauche,

```
// Static Objects
STATIC [
ROAD
TREE
]
// Dynamic Objects
DYNAMIC [
VEHICLE [
ID = vehicule_b;
INITDIRECTION = east;
```

Dialogue: The Persona Project from Microsoft Research

A conversation with Peedy

Turn	Utterance
	[Peedy is asleep on his perch]
User:	Good morning, Peedy.
	[Peedy rouses]
Peedy:	Good morning.
User:	Let's do a demo.
	[Peedy stands up, smiles]
Peedy:	Your wish is my command, what would you like to hear?
User:	What have you got by Bonnie Raitt?
	[Peedy waves in a stream of notes, and grabs one as they rush by.]
Peedy:	I have "The Bonnie Raitt Collection" from 1990.
User:	Pick something from that
Peedy:	How about "Angel from Montgomery"?

Dialogue: The Persona Project from Microsoft Research

User:	Sounds good.
	[Peedy drops note on pile]
Peedy:	OK.
User:	Play some rock after that.
	[Peedy scans the notes again, selects one]
Peedy:	How about "Fools in love"?
User:	Who wrote that?
	[Peedy cups one wing to his 'ear']
Peedy:	Huh?
User:	Who wrote that?
	[Peedy looks up, scrunches his brow]
Peedy:	Joe Jackson
User:	Fine.
	[Drops note on pile]

Persona System Architecture

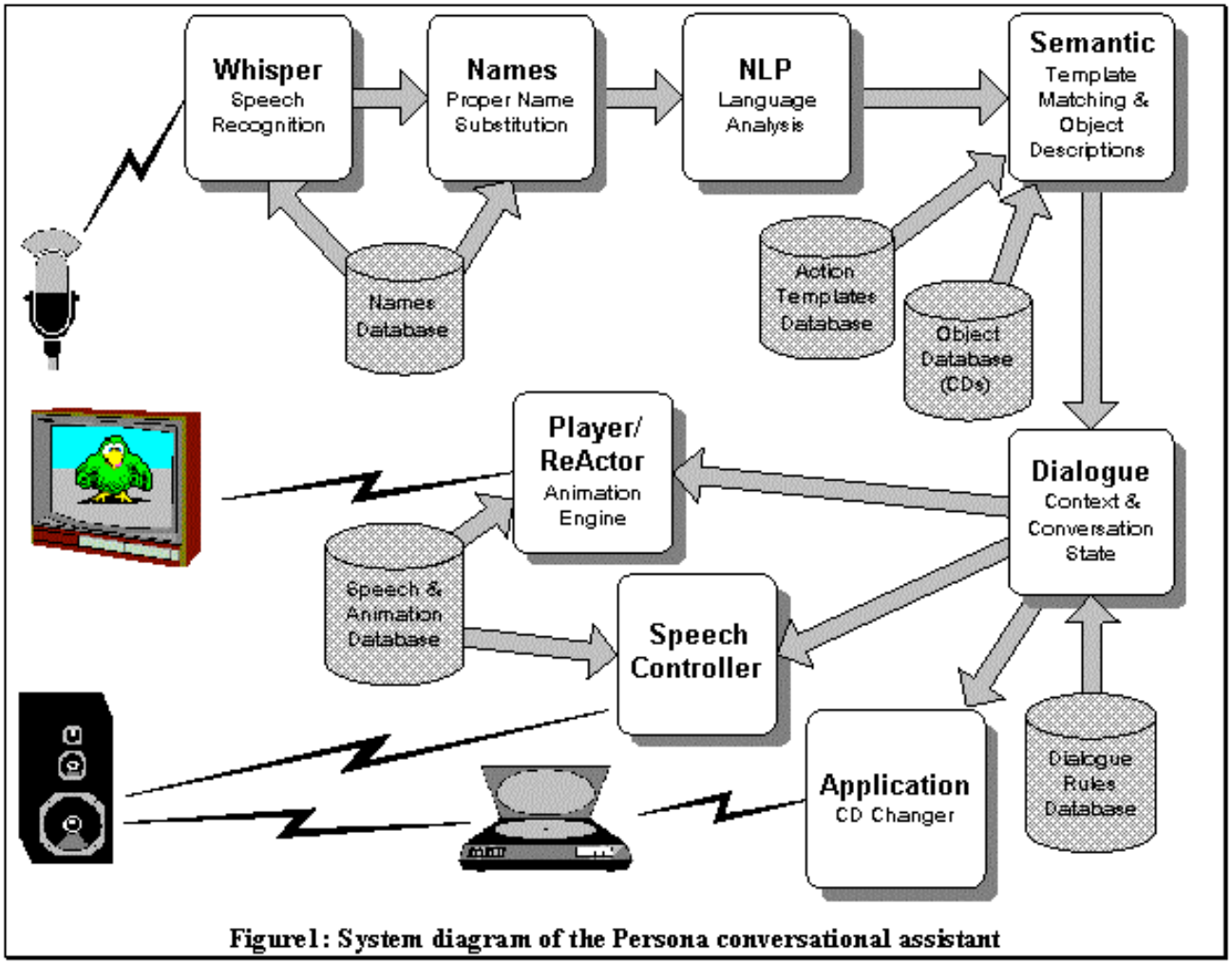


Figure1: System diagram of the Persona conversational assistant

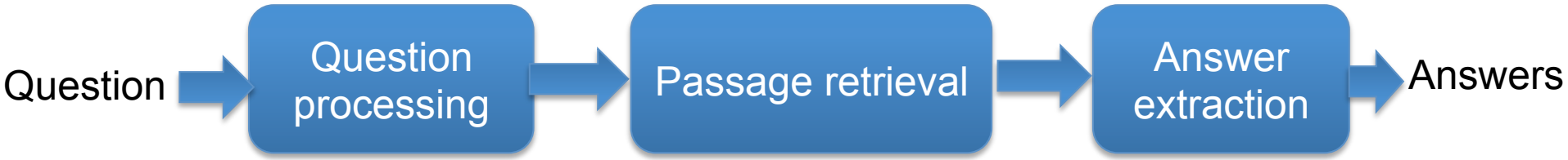
Source: http://research.microsoft.com/research/pubs/view.aspx?pubid=439

IBM Watson

- IBM Watson: A system that can answer questions better than any human
- Video: `https://www.youtube.com/watch?v=WFR3lOm_xhE`

- IBM Watson builds on the extraction of knowledge from masses of texts: Wikipedia, archive of the New York Times, etc.
- Bottom line: Text is the repository of human knowledge

IBM Watson: Simplified Architecture



Question parsing and classification:
Syntactic parsing, entity recognition, answer classification

Document retrieval. Extraction and ranking of passages:
Indexing, vector space model.

Extraction and ranking of answers:
Answer parsing, entity recognition

FMAN45
Maskininlärning
http://cs.lth.se/edan20/
Modern Techniques in Natural Language Processing

Pierre Nugues

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

May 23, 2019

Training Set

Part-of-speech taggers use a training set where every word is hand-annotated (Penn Treebank and CoNLL 2008).

Index	Word	Hand annotation	Index	Word	Hand annotation
1	Battle	JJ	19	of	IN
2	-	HYPH	20	their	PRP$
3	tested	JJ	21	countrymen	NNS
4	Japanese	JJ	22	to	TO
5	industrial	JJ	23	visit	VB
6	managers	NNS	24	Mexico	NNP
7	here	RB	25	,	,
8	always	RB	26	a	DT
9	buck	VBP	27	boatload	NN
10	up	RP	28	of	IN
11	nervous	JJ	29	samurai	FW
12	newcomers	NNS	30	warriors	NNS
13	with	IN	31	blown	VBN
14	the	DT	32	ashore	RB
15	tale	NN	33	375	CD
16	of	IN	34	years	NNS
17	the	DT	35	ago	RB
18	first	JJ	36	.	.

Part-of-Speech Tagging with Linear Classifiers

Linear classifiers are efficient devices to carry out part-of-speech tagging:

1. The lexical values are the input data to the tagger.
2. The parts of speech are assigned from left to right by the tagger.

Given the feature vector: $(w_{i-2}, w_{i-1}, w_i, w_{i+1}, w_{i+2}, t_{i-2}, t_{i-1})$, the classifier will predict the part-of-speech tag t_i at index i.

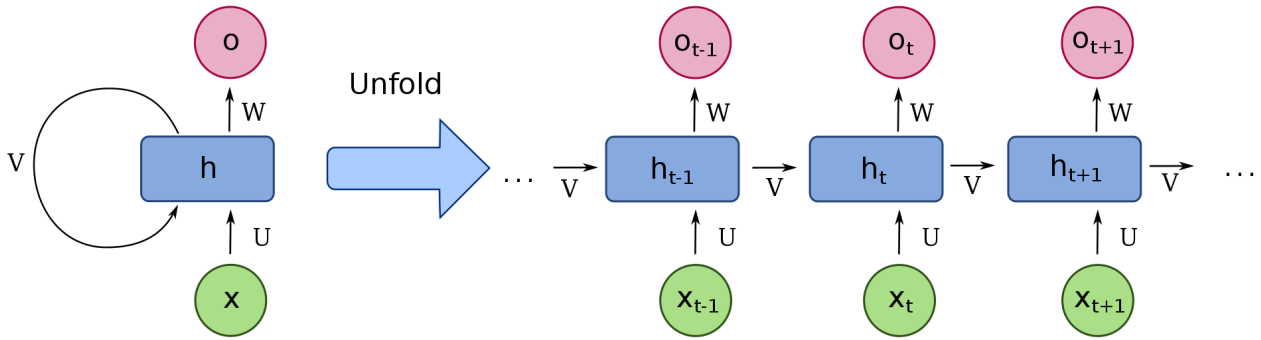
ID	FORM	PPOS	
	BOS	BOS	*Padding*
	BOS	BOS	
1	Battle	NN	
2	-	HYPH	
3	tested	NN	
...	...	...	
17	the	DT	
18	first	JJ	
19	of	IN	
20	their	PRP$	
21	countrymen	NNS	*Input features*
22	to	TO	
23	visit	VB	*Predicted tag*
24	Mexico		↓
25	,		
26	a		
27	boatload		
...	...	...	
34	years		
35	ago		
36	.		
	EOS		*Padding*
	EOS		

Feature Vectors

ID	Feature vectors							PPOS
	w_{i-2}	w_{i-1}	w_i	w_{i+1}	w_{i+2}	t_{i-2}	t_{i-1}	
1	BOS	BOS	Battle	-	tested	BOS	BOS	NN
2	BOS	Battle	-	tested	Japanese	BOS	NN	HYPH
3	Battle	-	tested	Japanese	industrial	NN	HYPH	JJ
...	...	...	...	...	...	...	...	...
19	the	first	of	their	countrymen	DT	JJ	IN
20	first	of	their	countrymen	to	JJ	IN	PRP$
21	of	their	countrymen	to	visit	IN	PRP$	NNS
22	their	countrymen	to	visit	Mexico	PRP$	NNS	TO
23	countrymen	to	visit	Mexico	,	NNS	TO	VB
24	to	visit	Mexico	,	a	TO	VB	NNP
25	visit	Mexico	,	a	boatload	VB	NNP	,
...	...	...	...	...	...	...	...	...
34	ashore	375	years	ago	.	RB	CD	NNS
35	375	years	ago	.	EOS	CD	NNS	RB
36	years	ago	.	EOS	EOS	NNS	RB	.

Sequence-to-Sequence

Recurrent neural networks:

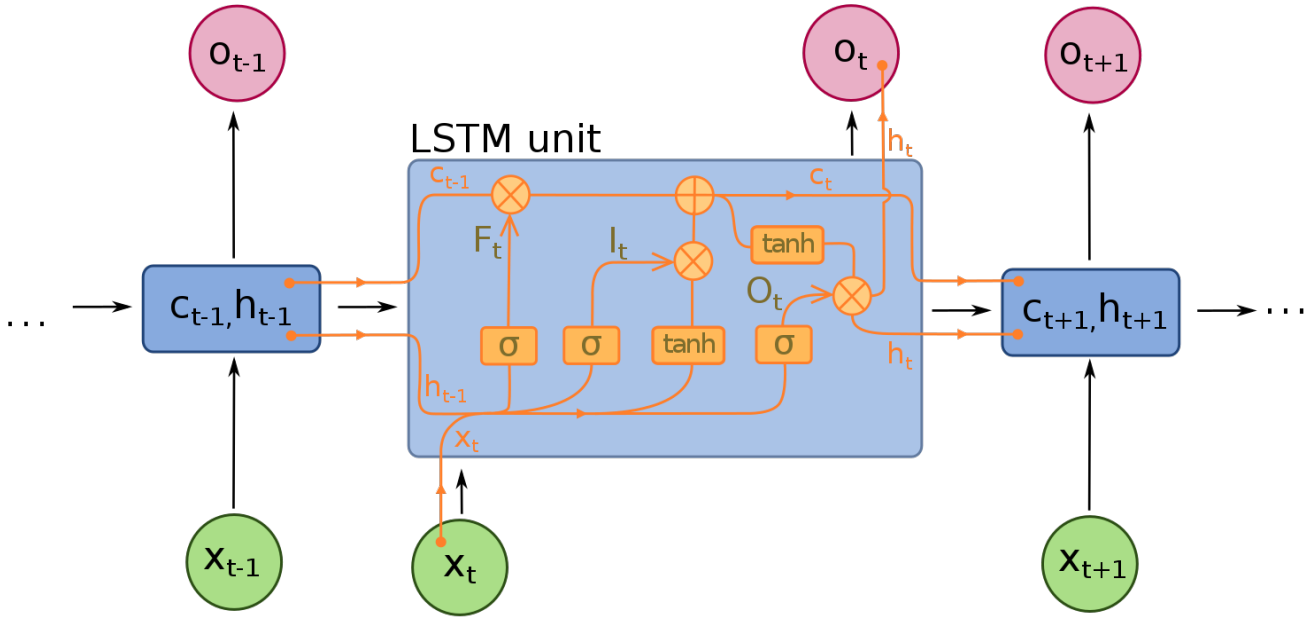


$$h_t = \sigma_h(Ux_t + Vh_{t-1} + b_h)$$
$$o_t = \sigma_o(Wh_t + b_o)$$

Source: Wikipedia

Sequence-to-Sequence

Long short-term memory (LSTM): add a cell state c as memory



Source: Wikipedia

Sequence-to-Sequence: The Equations

$$F_t = \sigma(W_F x_t + U_F h_{t-1} + b_F) \quad \text{(forget gate)} \quad (1)$$

$$I_t = \sigma(W_I x_t + U_I h_{t-1} + b_I) \quad \text{(input gate)} \quad (2)$$

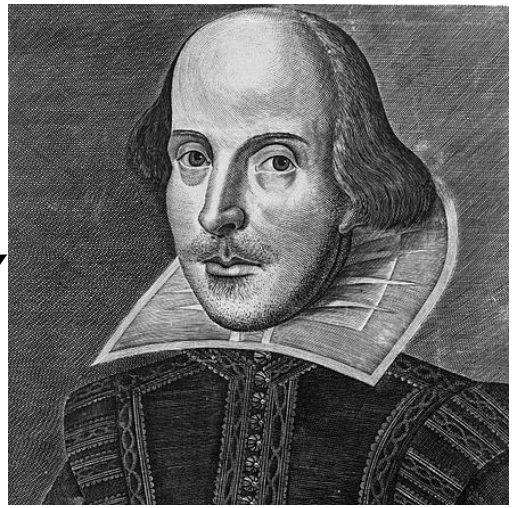
$$O_t = \sigma(W_O x_t + U_O h_{t-1} + b_O) \quad \text{(output gate)} \quad (3)$$

$$c_t = F_t \circ c_{t-1} + I_t \circ \tanh(W_c x_t + U_c h_{t-1} + b_c) \quad (4)$$

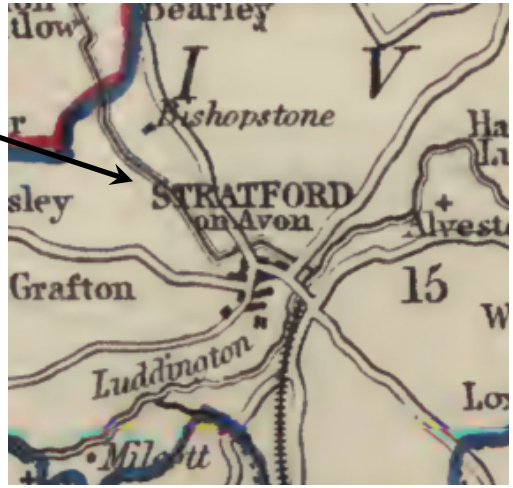
$$h_t = O_t \circ \tanh(c_t) \quad (5)$$

$$o_t = f(W_o h_t + b_o) \quad (6)$$

Named Entities: Proper Nouns



William Shakespeare was born and brought up in Stratford-upon-Avon



Others Entities: Common Nouns

Multiple Categories of Chunks

Extendable to any type of chunks: nominal, verbal, etc.
For the IOB scheme, this means tags such as I.Type, O.Type, and B.Type,
Types being NG, VG, PG, etc.
In CoNLL 2000, ten types of chunks

	Word	POS	Group		Word	POS	Group
	He	PRP	B-NP		*to*	TO	B-PP
	reckons	VBZ	B-VP		*only*	RB	B-NP
	the	DT	B-NP		*£*	#	I-NP
	current	JJ	I-NP		*1.8*	CD	I-NP
	account	NN	I-NP		*billion*	CD	I-NP
	deficit	NN	I-NP		*in*	IN	B-PP
	will	MD	B-VP		*September*	NNP	B-NP
	narrow	VB	I-VP		.	.	O

Noun groups (NP) are in red and verb groups (VP) are in blue.

IOB Annotation for Named Entities

CoNLL 2002	
Words	Named entities
Wolff	B-PER
,	O
currently	O
a	O
journalist	O
in	O
Argentina	B-LOC
,	O
played	O
with	O
Del	B-PER
Bosque	I-PER
in	O
the	O
final	O
years	O
of	O
the	O
seventies	O
in	O
Real	B-ORG
Madrid	I-ORG
.	O

CoNLL 2003			
Words	POS	Groups	Named entities
U.N.	NNP	I-NP	I-ORG
official	NN	I-NP	O
Ekeus	NNP	I-NP	I-PER
heads	VBZ	I-VP	O
for	IN	I-PP	O
Baghdad	NNP	I-NP	I-LOC
.	.	O	O

Information Extraction and Semantic Parsing

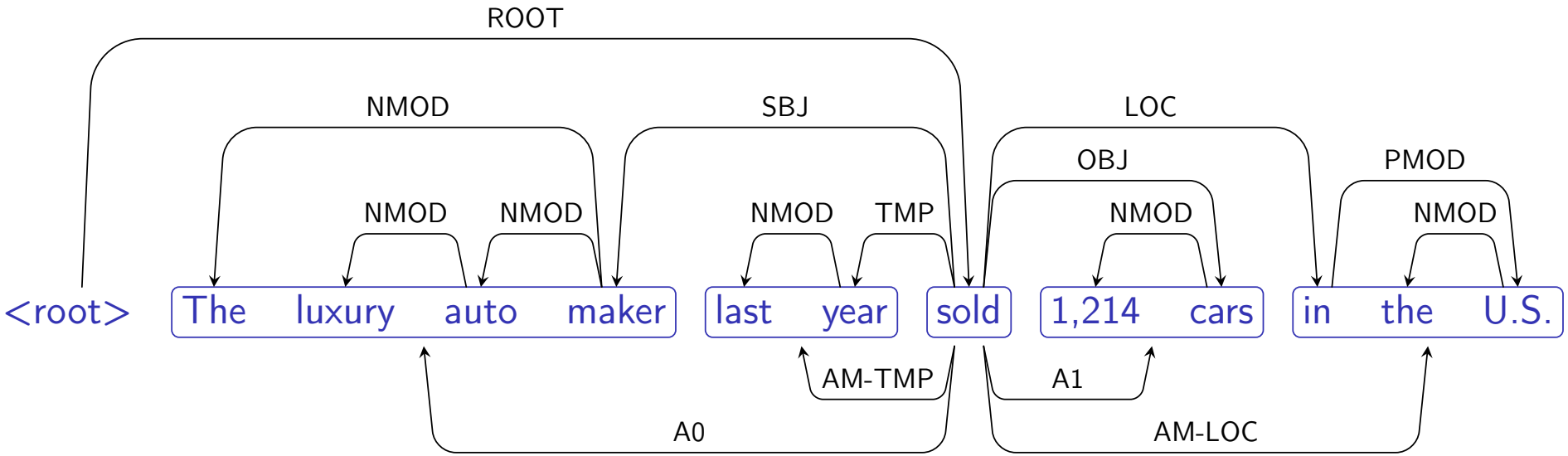
- Extract events (predicates) and their participants and circumstances (arguments) from text:
  Who, What, Where, When, Why?

Semantic dependencies (predicate–argument structures):

	He	crashed	his	car	into	a	tree	.
crash.01	A0		A1		AM-DIR			

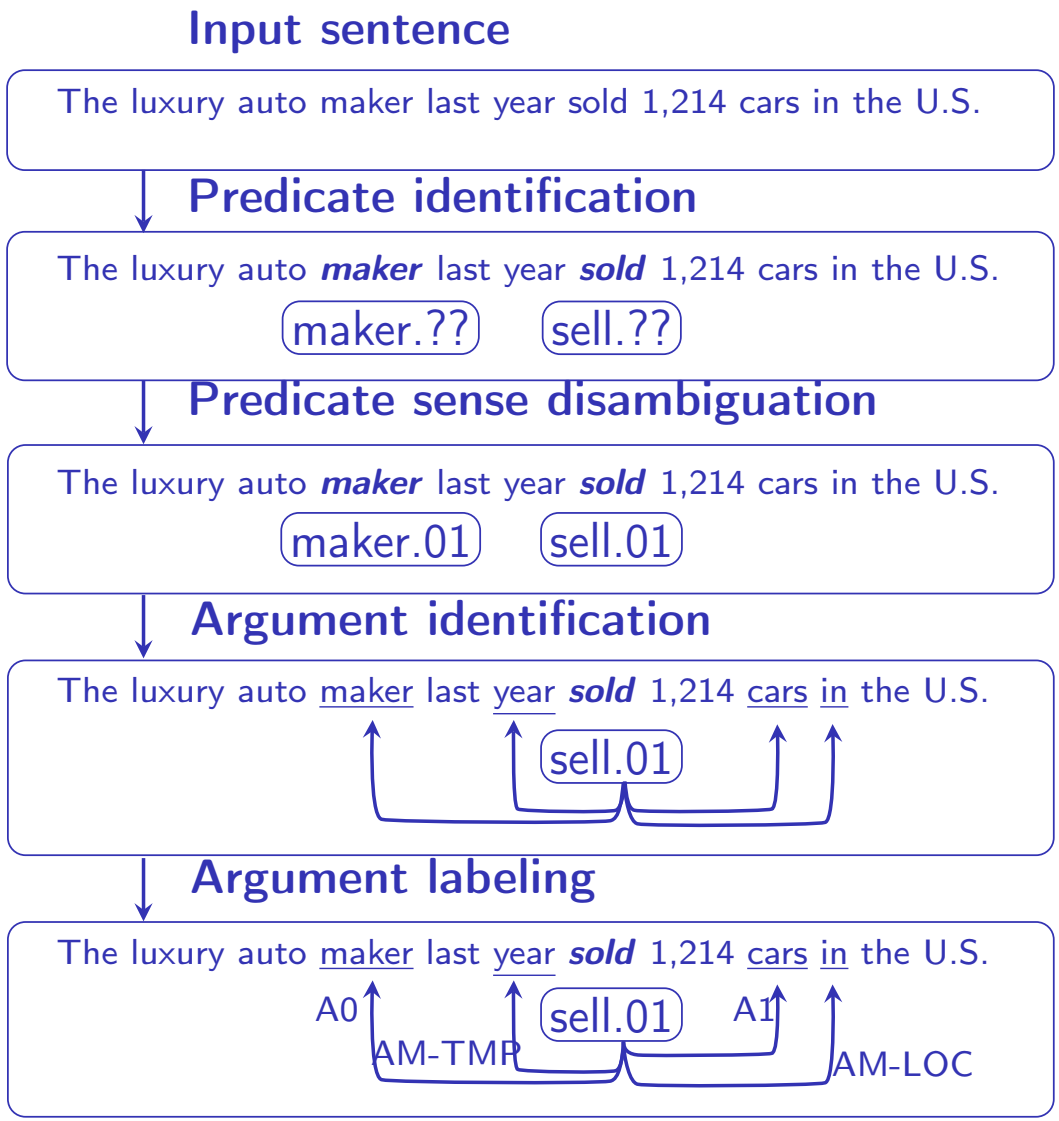
`http://barbar.cs.lth.se:8081/parse`

Alternative Visualization



	The	luxury	auto	maker	last	year	sold	1,214	cars	in	the	U.S.
maker.01		A1		A0								
sell.01	A0				AM-TMP			A1		AM-LOC		

Parsing Pipeline



Features for the Predicate Identification

Features used by Johansson and Nugues (2008) and values for *sold* in *The luxury auto maker last year sold 1,214 cars in the U.S.*

Feature	Value
PredForm	sold
PredLemma	sell
PredHeadForm	ROOT
PredHeadPOS	ROOT
PredDeprel	ROOT
ChildFormSet	{maker, year, cars, in}
ChildPOSSet	{NN, NNS, IN}
ChildDepSet	{SBJ, TMP, OBJ, LOC}
DepSubcat	SBJ+TMP+OBJ+LOC
ChildFormDepSet	{maker+SBJ, year+TMP, cars+OBJ, in+LOC}
ChildPOSDepSet	{NN+SBJ, NN+TMP, NNS+OBJ, IN+LOC}

Family of Classifiers

Supervised learning:

- Logistic regression
- Support vector machines
- Perceptron
- Neural networks

Modern techniques:

- Word embeddings
- LSTM

One Hot Encoding

- Bag of words is a frequent basic technique to represent text;
- It uses a one-hot encoding: A document is represented by a vector with as many dimensions (axes) as there are words in the collection;
- The coordinate on each word axis is set one when we have the word, else 0;
- A collection of two documents D1 and D2:

  D1: Chrysler plans new investments in Latin America.
  D2: Chrysler plans major investments in Mexico.

D.	america	chrysler	in	investments	latin	major	mexico	new	plans
1	1	1	1	1	1	0	0	1	1
2	0	1	1	1	0	1	1	0	1

- More elaborate representations use TFIDF.

Sparse Data

- In many tasks, models use word sequences;
- Current training sets rarely go beyond 1 million words;
- How do deal with sparse data, for instance:

Phrase	Occurrences on Google
"I have a class on Mondays"	6
"I have a class on Tuesdays"	5
"I have a class on Wednesdays"	6
"I have a class on Thursdays"	0
"I have a class on Fridays"	4

Table: Counts from Google

- In many other contexts, *Thursdays* has a behavior similar to the other days of the week;
- With one-hot encoding, it not closer to *Mondays* than to *class*.

Word Embeddings

- One-hot encoding is a discrete representation.
- Idea: Represent each word with a continuous vector of a reasonable dimension ∼300
- Two main techniques:
  - GloVe
  - word2vec

GloVe: Mutual Information

Mutual information, often called **pointwise mutual information**, defines an association strength between two words:

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

Often set to 0, when negative or the counts are 0.
`https://corpora.linguistik.uni-erlangen.de/cgi-bin/demos/Web1T5/Web1T5_colloc.perl`

	w_1	w_2	w_3	...	w_n
w_1	$C(w_1, w_1)$	$C(w_1, w_2)$	$C(w_1, w_3)$	...	$C(w_1, w_n)$
w_2	$C(w_2, w_1)$	$C(w_2, w_2)$	$C(w_2, w_3)$	...	$C(w_2, w_n)$
...					
w_n	$C(w_n, w_1)$	$C(w_n, w_2)$	$C(w_n, w_3)$	...	$C(w_n, w_n)$

A sparse matrix

Documents

Possible to consider documents instead:

Words\Docs	D_1	D_2	D_3	...	D_n
w_1	$C(w_1, D_1)$	$C(w_1, D_2)$	$C(w_1, D_3)$	...	$C(w_1, D_n)$
w_2	$C(w_2, D_1)$	$C(w_2, D_2)$	$C(w_2, D_3)$	...	$C(w_2, D_n)$
...					
w_m	$C(w_m, D_1)$	$C(w_m, D_2)$	$C(w_m, D_3)$	...	$C(w_m, D_n)$

Principal Component Analysis

We can carry out a single value decomposition of $\mathbf{X}$:

$$\mathbf{X} = \mathbf{U}\Sigma\mathbf{V},$$

where $\mathbf{X}$ is the cooccurrence matrix.

- $\mathbf{U}$, a $m \times m$ matrix;
- Σ, a $m \times n$ diagonal matrix;
- $\mathbf{V}$, a $n \times n$ matrix;

The projections of the words in the 300 first dimensions:
$\mathbf{U}'\Sigma'$, (or simply $\mathbf{U}'$), a $m \times 300$ matrix;
Easy to program, but does not scale to very large corpora.

Glove

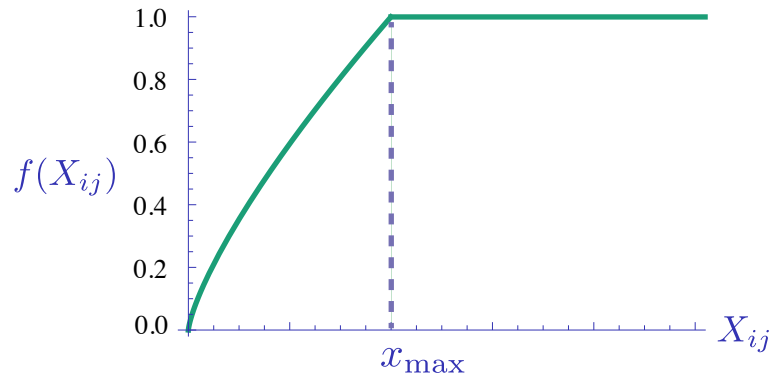
Try to decompose:

$$\begin{aligned} \mathbf{X} &= \mathbf{W}\mathbf{W}' + \vec{b} + \vec{b}' \\ \log x_{ij} &= \vec{w}_i^T \vec{w}'_j + b_i + b'_j \end{aligned}$$

where $\vec{w}$ has 300 dimensions.
We define an objective function:

$$J = \sum_{i=1}^{V} \sum_{j=1}^{V} f\left(x_{ij}\right) \left(\vec{w}_i^T \vec{w}'_j + b_i + b'_j - \log x_{ij}\right)^2$$

$$f\left(x_{ij}\right) = \begin{cases} \left(\frac{x_{ij}}{x_{\max}}\right)^{\alpha} & \text{if } x_{ij} < x_{\max} \\ 1 & \text{otherwise.} \end{cases}$$



Implementation: `http://www.foldl.me/2014/glove-python/`

Word2vec

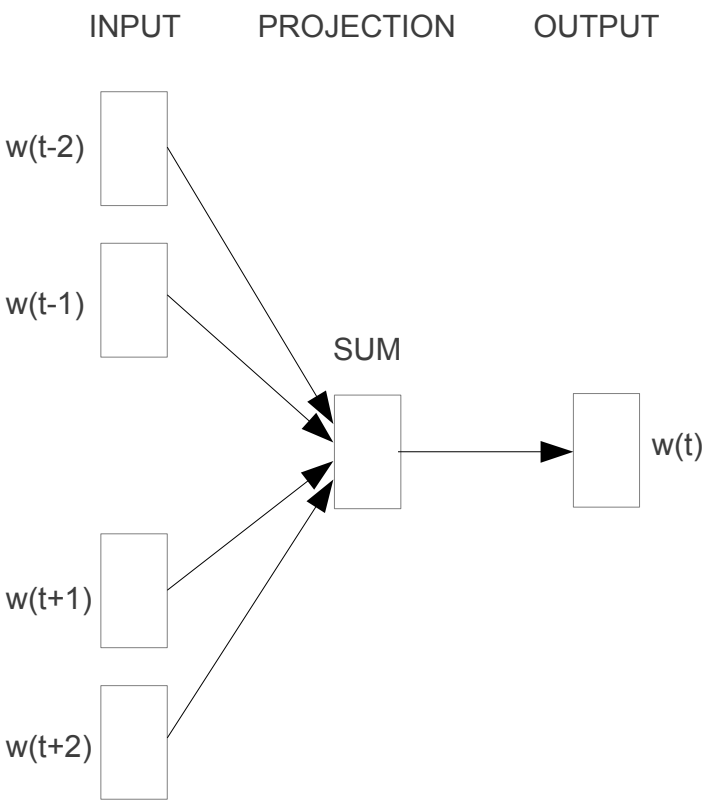
*Chrysler plans new **investments** in Latin America*

Two viewpoints:

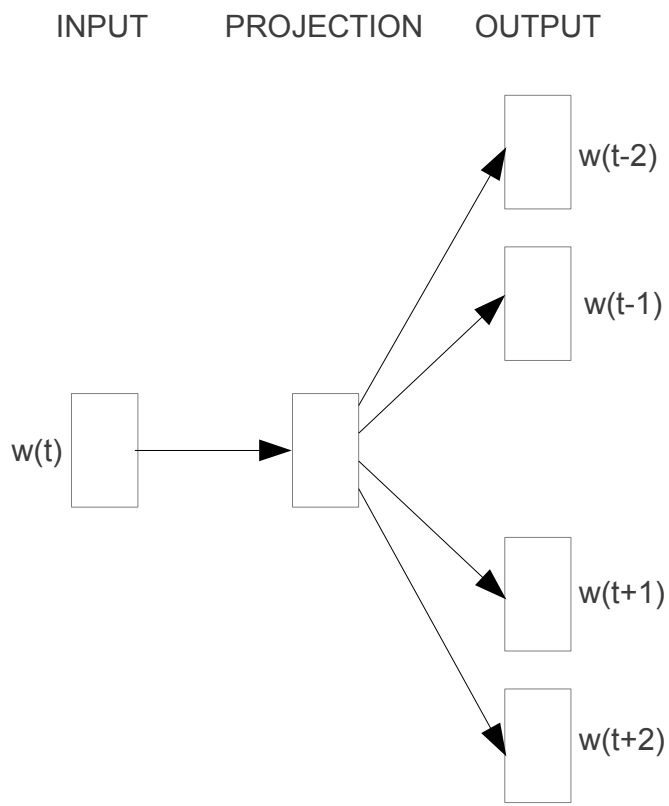
- Skipgrams: Predict a word from its context, for instance *investments* from
  Chrysler plans new [???] in Latin America
- CBOW: Predict the context distribution of a word:
  *[???] [???] [???] **investments** [???] [???] [???]*

Typically 5-grams (sequences of 5 words)

Skipgrams and CBOW



CBOW



Skip-gram

From Mikolov et al. Efficient Estimation of Word Representations in Vector Space, 2013.

CBOW

Chrysler plans new ***investments*** *in Latin America*

$$\text{Maximize } P(\text{word}|\text{context})$$
$$\arg\max_w P(w|w_{i-2}, w_{i-1}, w_{i+1}, w_{i+2})$$
$$\arg\max_w P(w|w_{i-2} = \text{plans}, w_{i-1} = \text{new}, w_{i+1} = \text{in}, w_{i+2} = \text{Latin})$$

We make the approximations:

$$\arg\max_w \prod_{j=-2, j\neq 0}^{2} P(w|w_{i+j})$$
$$\arg\max_w \prod_{cw \in \{\text{plans, new, in, Latin}\}} P(w|cw)$$

This for all the words in the vocabulary V, so that given a context, you find the word and rank the other candidates.

CBOW: Moving to a Continuous Space

Moving to a continuous space, where $\mathbf{u}$ and $\mathbf{v}$ are vectors of, say, 300 dimensions;
The similarity of two vectors is often measured by the cosine similarity or dot product:

$$\mathbf{u} \cdot \mathbf{v}$$

If $\mathbf{u}$ is the representation of *investments*, and $\mathbf{v}$, the representation of a similar word, they should have a high cosine similarity;
As a trick, we use the generalized logistic function, softmax, to translate this in a probability:

$$P_{\mathbf{u}_i}(\mathbf{x}) = \frac{e^{\mathbf{u}_i \cdot \mathbf{x}}}{\sum_{j=1}^{|V|} e^{\mathbf{u}_j \cdot \mathbf{x}}}$$

where we denote $\mathbf{u}_i$ the representation of the word v_i in the vocabulary V.

CBOW

This for all the words in the vocabulary V:

$$\arg\max \prod_{w_i \in V} \prod_{j=-2, j\neq 0}^{2} P(w_{i+j}|w_i)$$

CBOW:Representation

- Input: the context, four one-hot vectors, here *major*, dimension $|V|$, where V is the vocabulary;
- The input embedding layer **IE**, matrix of dimension $D \times |V|$, shared between the input words, where D is the embedding dimension, we will take the mean of the context words;
- The output embeddings **OE**, dimension $|V| \times D$.
- Output layer, one one-hot vector, dimension $|V| \times 1$, This layer uses a softmax activation layer.

$$\mathbf{w_i} = \begin{bmatrix}0\\0\\0\\0\\0\\1\\0\\0\\0\end{bmatrix}; \mathbf{IE} = \begin{bmatrix}\dots & \dots & \dots & \dots & \dots & 0.3 & \dots & \dots & \dots\\ \dots & \dots & \dots & \dots & \dots & 0.7 & \dots & \dots & \dots\\ \dots & \dots & \dots & \dots & \dots & -0.5 & \dots & \dots & \dots\\ \dots & \dots & \dots & \dots & \dots & 0.4 & \dots & \dots & \dots\end{bmatrix}; \mathbf{OE} = \begin{bmatrix}\dots & \dots & \dots & \dots\\ \dots & \dots & \dots & \dots\\ \dots & \dots & \dots & \dots\\ \dots & \dots & \dots & \dots\\ \dots & \dots & \dots & \dots\\ \dots & \dots & \dots & \dots\\ \dots & \dots & \dots & \dots\\ \dots & \dots & \dots & \dots\\ \dots & \dots & \dots & \dots\end{bmatrix}; \mathbf{w_o} = \begin{bmatrix}0\\0\\0\\1\\0\\0\\0\\0\\0\end{bmatrix}$$

Translation

Given a sentence, a phrase, or a word in a **source language**, find its equivalent in the **target language**. Traditional notation, E(nglish) and F(rench)
Statistical machine translation maximizes:

$$\begin{aligned} E = & \arg\max P(E|F) \\ = & \arg\max P(F|E)P(E) \end{aligned}$$

- $P(E)$ is estimated using a language model
- Most machine translation techniques use parallel corpora to compute $P(F|E)$

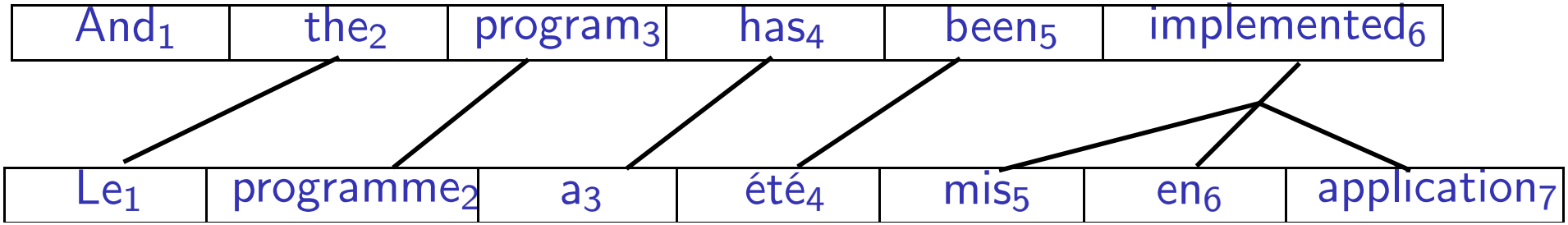
Generate pairs of sentences from the target and source texts, assigns them a score, which corresponds to the difference of lengths in characters of the aligned pairs,
Uses dynamic programming to find the maximum likelihood alignment of sentences.

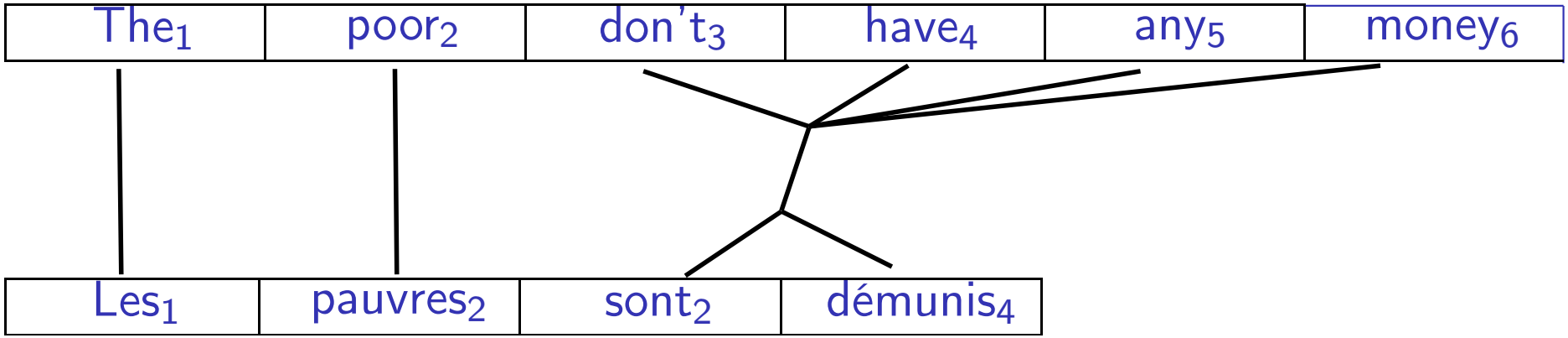
Parallel Corpora (Swiss Federal Law)

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 beeinträchtigen können.	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.
2 Wird Milch ausserhalb	2 Si le lait destiné à	2 Se viene collocato fuori

Alignment (Brown et al. 1993)

Canadian Hansard





Sequence-to-Sequence

Most accurate techniques use recurrent networks with parallel corpora.

- RNN
- LSTM

Delightful implementation in Keras: `https://blog.keras.io/a-ten-minute-introduction-to-sequence-to-sequence-learning-in-html`

Big jump in performance the last three years