L90: Overview of Natural Language Processing

Lecture 1: Overview of Overview of Natural Language Processing

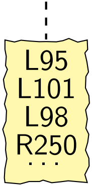
Simone Teufel

Department of Computer Science and Technology
University of Cambridge

Michaelmas 2021/22

L90: Overview of Natural Language Processing

L90: Overview of Natural Language Processing



L95
L101
L98
R250
. . .

breadth over depth

L90: Overview of Natural Language Processing

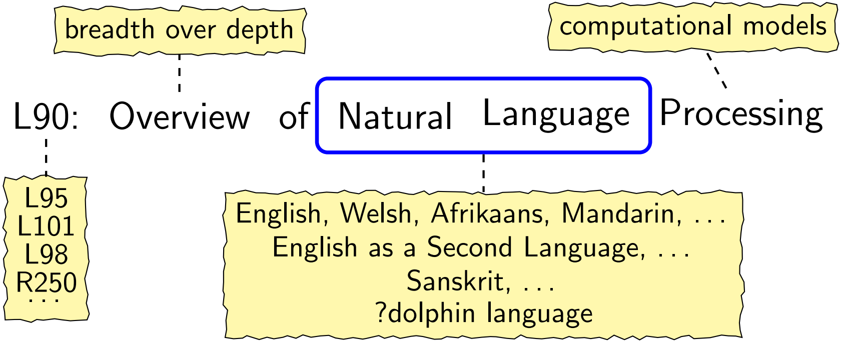
L95
L101
L98
R250
...

breadth over depth

L90: Overview of Natural Language Processing

L95
L101
L98
R250
...

English, Welsh, Afrikaans, Mandarin, ...
English as a Second Language, ...
Sanskrit, ...
?dolphin language



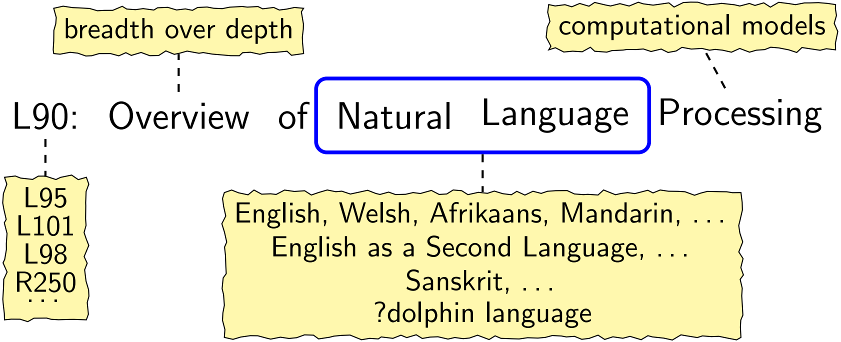
breadth over depth

computational models

L90: Overview of Natural Language Processing

L95
L101
L98
R250
...

English, Welsh, Afrikaans, Mandarin, ...
English as a Second Language, ...
Sanskrit, ...
?dolphin language



breadth over depth

computational models

L90: Overview of Natural Language Processing

L95
L101
L98
R250
...

English, Welsh, Afrikaans, Mandarin, ...
English as a Second Language, ...
Sanskrit, ...
?dolphin language

Lecture 1: Overview of Overview of Natural Language Processing

1. What does it mean to know a language?
2. Form transformation
3. Why NLP is hard?

based on
Weiwei Sun's and
Ann Copestake's
previous lecture slides

What does it mean to know a language?

universal translator

Hoshi Sato Explaining the Universal Translator to School Children

▶ `www.youtube.com/watch?v=wtAmPX1Itr0`

What does it mean to *know* a language?

Some yinkish dripners blorked quastofically into the nindin with the pidibs.

the example is partly from A Carnie's *Syntax: A Generative Introduction*

What does it mean to *know* a language?

Some yinkish dripners blorked quastofically into the nindin with the pidibs.

the example is partly from A Carnie's *Syntax: A Generative Introduction*

- there was a BLORK event;

What does it mean to *know* a language?

Some yinkish dripners blorked quastofically into the nindin with the pidibs.

the example is partly from A Carnie's *Syntax: A Generative Introduction*

- there was a BLORK event;
- it happened in the PAST;

What does it mean to *know* a language?

Some yinkish dripners blorked quastofically into the nindin with the pidibs.

the example is partly from A Carnie's *Syntax: A Generative Introduction*

- there was a BLORK event;
- it happened in the PAST;
- the AGENT of BLORK is dripners;

What does it mean to *know* a language?

Some yinkish dripners blorked quastofically into the nindin with the pidibs.

the example is partly from A Carnie's *Syntax: A Generative Introduction*

- there was a BLORK event;
- it happened in the PAST;
- the AGENT of BLORK is dripners;
- the dripners were YINKISH;

What does it mean to *know* a language?

Some yinkish dripners blorked quastofically into the nindin with the pidibs.

the example is partly from A Carnie's *Syntax: A Generative Introduction*

- there was a BLORK event;
- it happened in the PAST;
- the AGENT of BLORK is dripners;
- the dripners were YINKISH;
- SOME but NOT ALL dripners blorked;

What does it mean to *know* a language?

Some yinkish dripners blorked quastofically into the nindin with the pidibs.

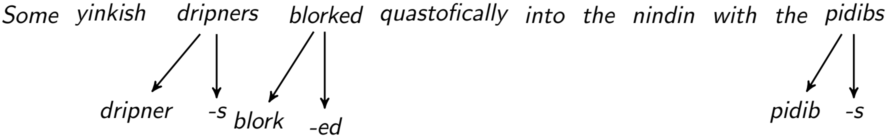
the example is partly from A Carnie's *Syntax: A Generative Introduction*

- there was a BLORK event;
- it happened in the PAST;
- the AGENT of BLORK is dripners;
- the dripners were YINKISH;
- SOME but NOT ALL dripners blorked;
- WITH THE PIDIBS may talk about NINDIN or BLORK;

Structuring a sentence

Some yinkish dripners blorked quastofically into the nindin with the pidibs

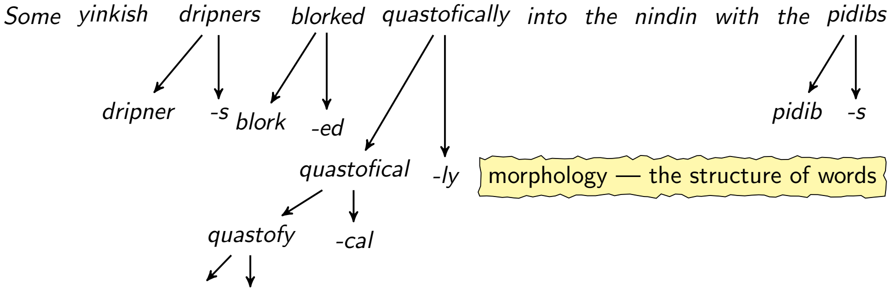
Structuring a sentence



Some yinkish dripners blorked quastofically into the nindin with the pidibs

dripner -s blork -ed pidib -s

Structuring a sentence



Some yinkish dripners blorked quastofically into the nindin with the pidibs

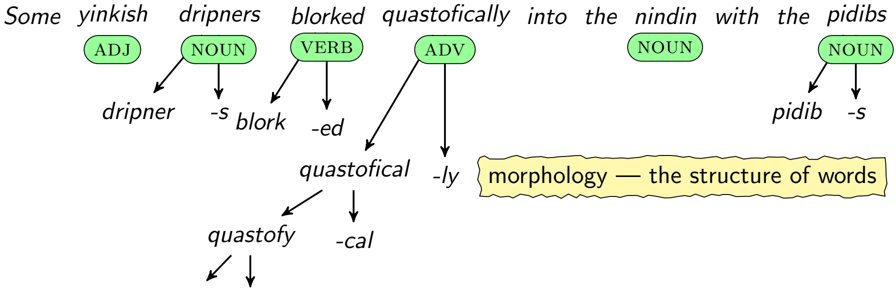
dripner *-s* *blork* *-ed* *pidib* *-s*

quastofical *-ly*

quastofy *-cal*

morphology — the structure of words

Structuring a sentence



Some yinkish dripners blorked quastofically into the nindin with the pidibs

ADJ NOUN VERB ADV NOUN NOUN

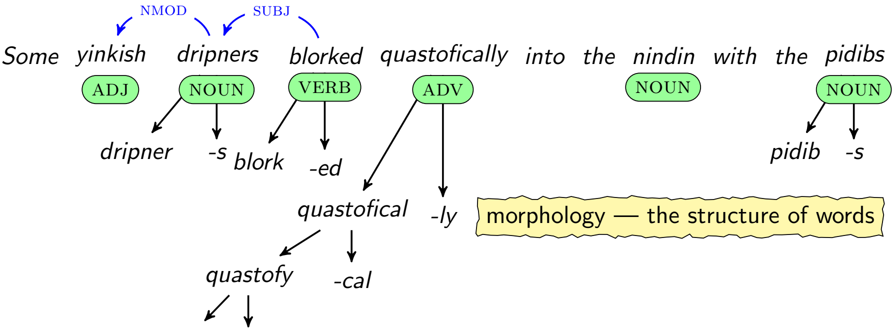
dripner -s blork -ed pidib -s

quastofical -ly

quastofy -cal

morphology — the structure of words

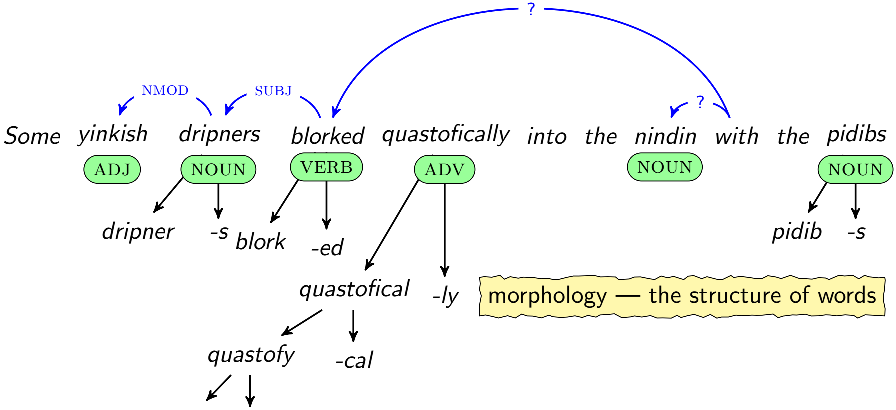
Structuring a sentence



Some yinkish dripners blorked quastofically into the nindin with the pidibs

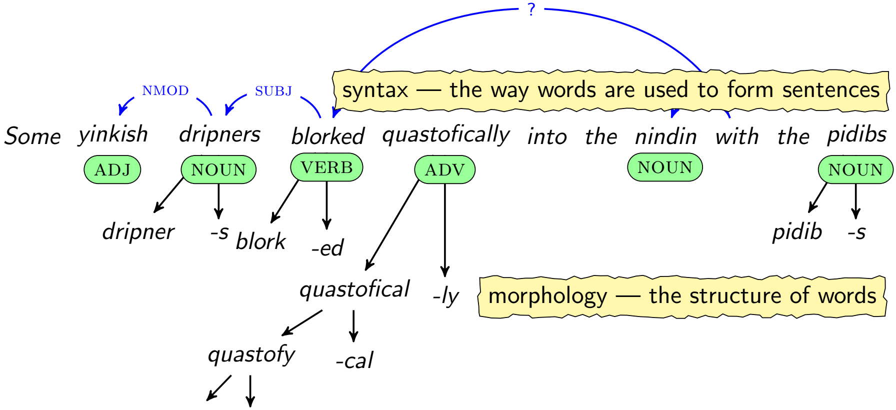
morphology — the structure of words

Structuring a sentence



morphology — the structure of words

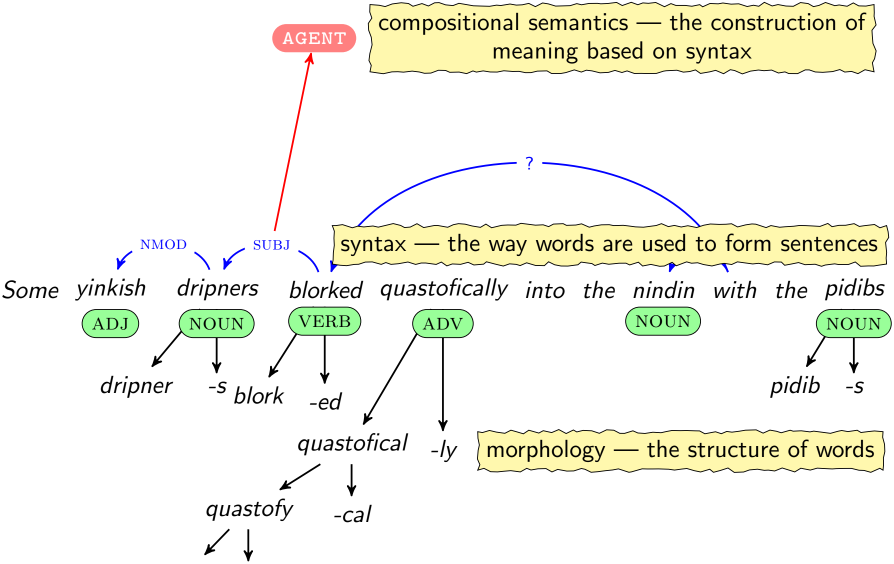
Structuring a sentence



syntax — the way words are used to form sentences

morphology — the structure of words

Structuring a sentence



compositional semantics — the construction of meaning based on syntax

syntax — the way words are used to form sentences

morphology — the structure of words

AGENT

?

NMOD

SUBJ

Some yinkish dripners blorked quastofically into the nindin with the pidibs

ADJ

NOUN

VERB

ADV

NOUN

NOUN

dripner *-s*

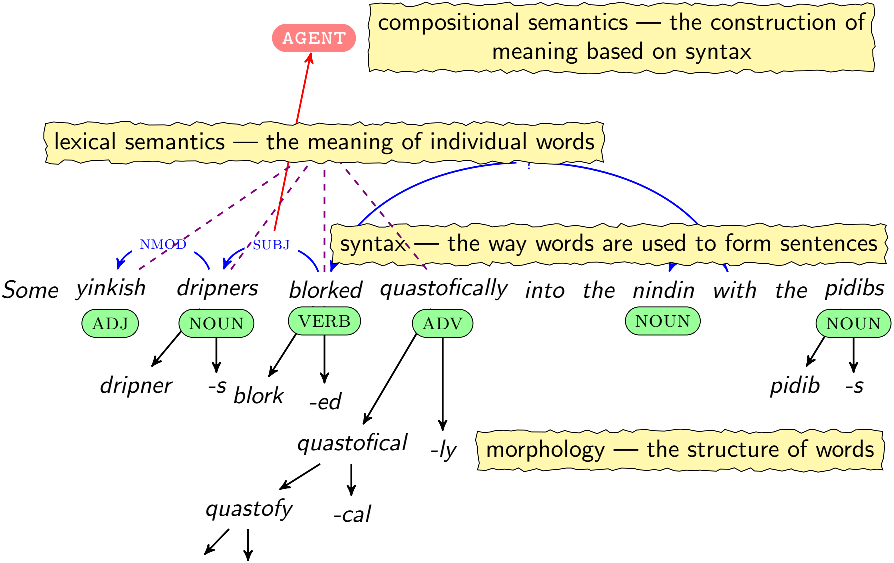
blork *-ed*

quastofical *-ly*

quastofy *-cal*

pidib *-s*

Structuring a sentence



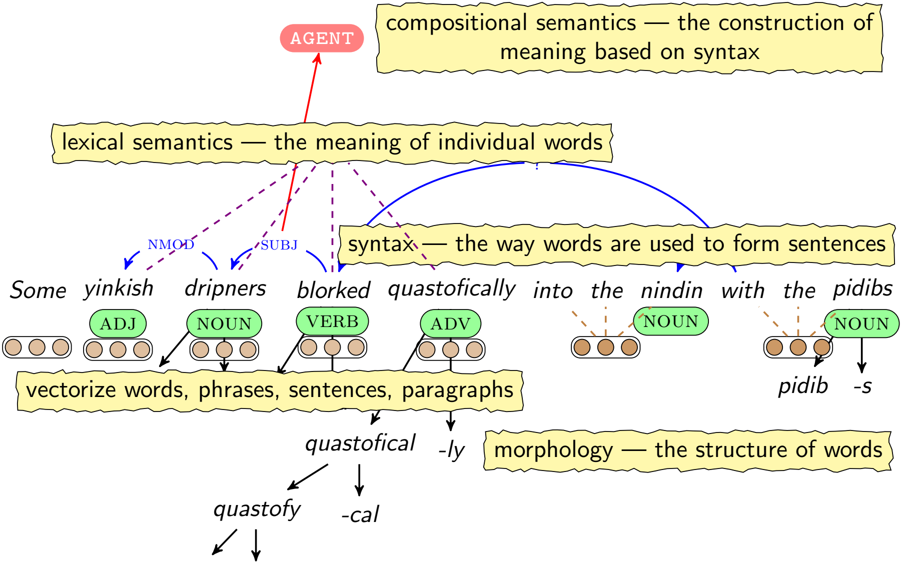
compositional semantics — the construction of meaning based on syntax

lexical semantics — the meaning of individual words

syntax — the way words are used to form sentences

morphology — the structure of words

Structuring a sentence



compositional semantics — the construction of meaning based on syntax

lexical semantics — the meaning of individual words

syntax — the way words are used to form sentences

vectorize words, phrases, sentences, paragraphs

morphology — the structure of words

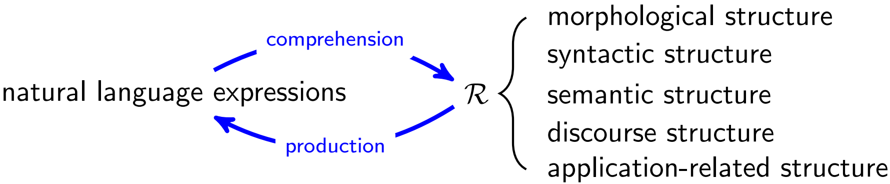
NLP: the computational modelling of human language

- *Morphology* — the structure of words: lecture 2.
- *Syntax* — the way words are used to form phrases: lectures 3, 5 and 6.
- *Semantics*
  - *Compositional semantics* — the construction of meaning based on syntax: lecture 9.
  - *Lexical semantics* — the meaning of individual words: lecture 8 (sort of) and 10.
- *Pragmatics* — meaning in context: lecture 11.
- *Language generation* — lecture 12.

What model?

- *Symbolic models* — finite-state machines and context-free grammars: lecture 2 and 5.
- *Statistical models* — classification: lecture 3.
- *Neural models* — (sequential) classification: lecture 4 and 7.

Form transformation



natural language expressions

comprehension

production

$\mathcal{R}$

morphological structure

syntactic structure

semantic structure

discourse structure

application-related structure

CoNLL shared tasks

- The SIGNLL Conference on Computational Natural Language Learning
- https://www.conll.org/previous-tasks

2020/2019	Cross-Framework Meaning Representation Parsing
2018/2017	Multilingual Parsing from Raw Text to Universal Dependencies
2018/2017	Universal Morphological Reinflection
2016/2016	(Multilingual) Shallow Discourse Parsing
2014/2013	Grammatical Error Correction
2012/2011	Modelling (Multilingual) Unrestricted Coreference in OntoNotes
2010	Hedge Detection
2009/2008	Syntactic and Semantic Dependencies in English/Multiple Languages
2007/2006	Multi-Lingual Dependency Parsing (Domain Adaptation)
2005/2004	Semantic Role Labeling
2003/2002	Language-Independent Named Entity Recognition
2001	Clause Identification
2000	Chunking
1999	NP Bracketing

CoNLL ST 1999/2000/2002/2003/2006/2007/2017/2018

input words

1	`U.N.`
2	`official`
3	`Ekeus`
4	`heads`
5	`for`
6	`Baghdad`
7	`.`

CoNLL ST 1999/2000/2002/2003/2006/2007/2017/2018

part-of-speech tags

input words

1	U.N.	NNP
2	official	NN
3	Ekeus	NNP
4	heads	VBZ
5	for	IN
6	Baghdad	NNP
7	.	.

CoNLL ST 1999/2000/2002/2003/2006/2007/2017/2018

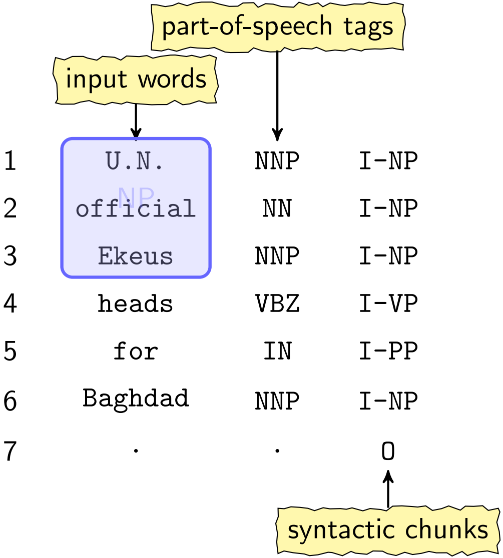
part-of-speech tags

input words

1	U.N.	NNP	I-NP
2	official	NN	I-NP
3	Ekeus	NNP	I-NP
4	heads	VBZ	I-VP
5	for	IN	I-PP
6	Baghdad	NNP	I-NP
7	.	.	O

syntactic chunks

CoNLL ST 1999/2000/2002/2003/2006/2007/2017/2018



part-of-speech tags

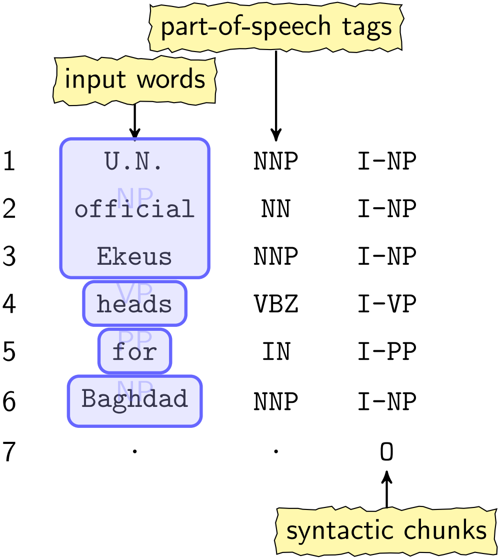
input words

1	U.N.	NNP	I-NP
2	official	NN	I-NP
3	Ekeus	NNP	I-NP
4	heads	VBZ	I-VP
5	for	IN	I-PP
6	Baghdad	NNP	I-NP
7	.	.	O

NP

syntactic chunks

CoNLL ST 1999/2000/2002/2003/2006/2007/2017/2018



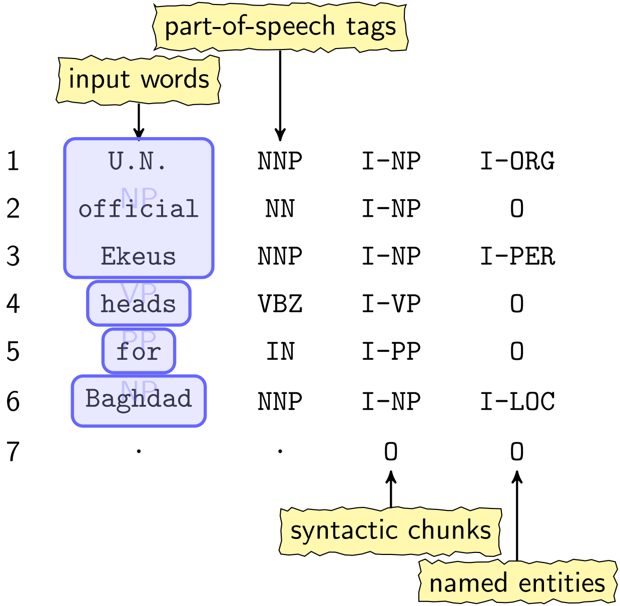
part-of-speech tags

input words

1	U.N.	NNP	I-NP
2	official	NN	I-NP
3	Ekeus	NNP	I-NP
4	heads	VBZ	I-VP
5	for	IN	I-PP
6	Baghdad	NNP	I-NP
7	.	.	O

syntactic chunks

CoNLL ST 1999/2000/2002/2003/2006/2007/2017/2018



part-of-speech tags

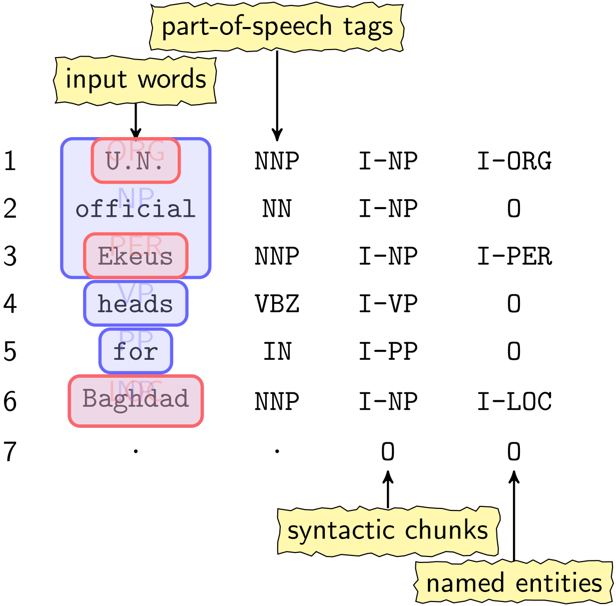
input words

1	U.N.	NNP	I-NP	I-ORG
2	official	NN	I-NP	O
3	Ekeus	NNP	I-NP	I-PER
4	heads	VBZ	I-VP	O
5	for	IN	I-PP	O
6	Baghdad	NNP	I-NP	I-LOC
7	.	.	O	O

syntactic chunks

named entities

CoNLL ST 1999/2000/2002/2003/2006/2007/2017/2018



part-of-speech tags

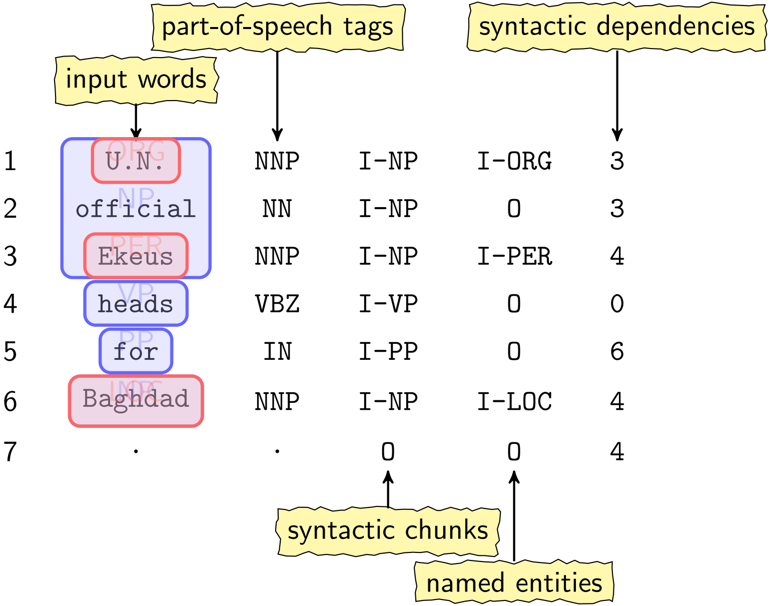
input words

1	U.N.	NNP	I-NP	I-ORG
2	official	NN	I-NP	O
3	Ekeus	NNP	I-NP	I-PER
4	heads	VBZ	I-VP	O
5	for	IN	I-PP	O
6	Baghdad	NNP	I-NP	I-LOC
7	.	.	O	O

syntactic chunks

named entities

CoNLL ST 1999/2000/2002/2003/2006/2007/2017/2018



1	U.N.	NNP	I-NP	I-ORG	3
2	official	NN	I-NP	O	3
3	Ekeus	NNP	I-NP	I-PER	4
4	heads	VBZ	I-VP	O	0
5	for	IN	I-PP	O	6
6	Baghdad	NNP	I-NP	I-LOC	4
7	.	.	O	O	4

input words

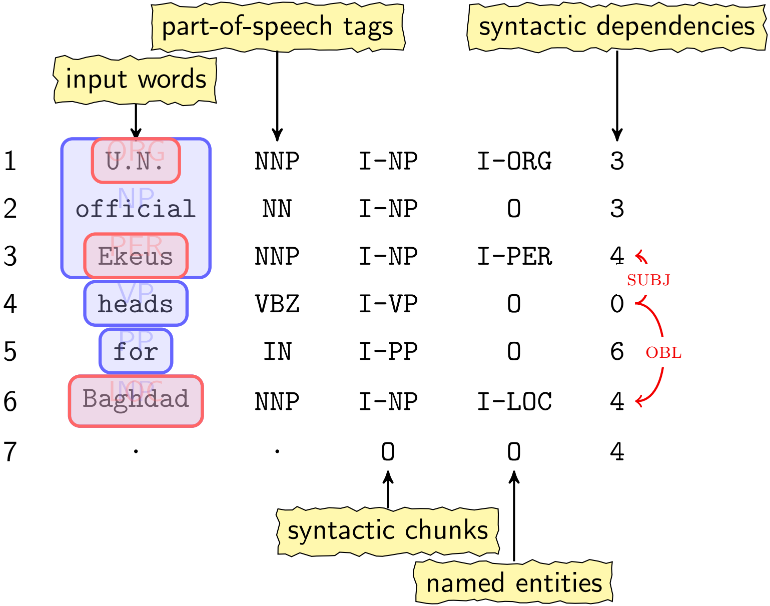
part-of-speech tags

syntactic dependencies

syntactic chunks

named entities

CoNLL ST 1999/2000/2002/2003/2006/2007/2017/2018



input words

part-of-speech tags

syntactic dependencies

1	U.N.	NNP	I-NP	I-ORG	3
2	official	NN	I-NP	O	3
3	Ekeus	NNP	I-NP	I-PER	4
4	heads	VBZ	I-VP	O	0
5	for	IN	I-PP	O	6
6	Baghdad	NNP	I-NP	I-LOC	4
7	.	.	O	O	4

syntactic chunks

named entities

Instructions

Natural language: *Go to the third junction and take a left*

Programming language:

```
(do-seq(do-n-times 3
  (move-to forward-loc
    (do-until
      (junction current-loc
        (move-to forward-loc))))
  (turn-right))
```

Instructions

Natural language: *Go to the third junction and take a left*

Programming language:

```
(do-seq(do-n-times 3
  (move-to forward-loc
    (do-until
      (junction current-loc
        (move-to forward-loc))))
  (turn-right))
```

Many other application-based representations

P Dasigi, S Iyer, A Suhr, M Gardner and L Zettlemoyer. ACL 2018 tutorial on neural semantic parsing.

https://github.com/allenai/acl2018-semantic-parsing-tutorial/

Querying a knowledge base

User query: *Has my order number 4291 been shipped yet?*

Database:

Order number	**Date ordered**	**Date shipped**
4290	2/2/13	2/2/13
4291	2/2/13	2/2/13
4292	2/2/13	

Conversational User Interface

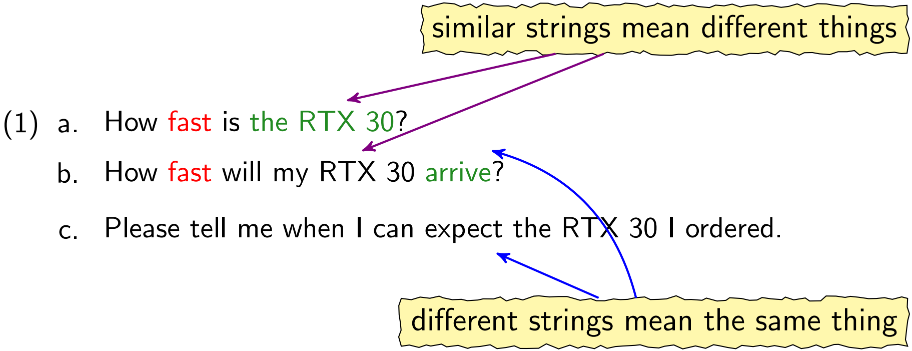
USER: *Has my order number 4291 been shipped yet?*

DB QUERY: `order(number=4291,date_shipped=?)` $\triangleright\mathcal{R}$

RESPONSE: *Order number 4291 was shipped on 2/2/13*

Why NLP is hard?

Why is this difficult?



similar strings mean different things

(1) a. How fast is the RTX 30?

b. How fast will my RTX 30 arrive?

c. Please tell me when I can expect the RTX 30 I ordered.

different strings mean the same thing

Why is this difficult?

(2) a. Do you sell Sony laptops and disk drives?

b. Do you sell (Sony (laptops and disk drives))?

c. Do you sell (Sony laptops) and (disk drives)?

ambiguity

$\boxed{2 \times (3+4) = 2 \times 3 + 2 \times 4}$ vs $\boxed{2 \times 3 + 4}$

Wouldn't it be better if . . . ?

The properties which make natural language difficult to process are essential to human communication:

- Flexible
- Learnable but compact
- Emergent, evolving systems

Synonymy and ambiguity go along with these properties.

Natural language communication can be indefinitely precise:

- Ambiguity is mostly local (for humans)

General comments

- Even *simple* applications might need complex knowledge sources.

General comments

- Even *simple* applications might need complex knowledge sources.
- Applications cannot be 100% perfect.

General comments

- Even *simple* applications might need complex knowledge sources.
- Applications cannot be 100% perfect.
- Applications that are $< 100\%$ perfect can be useful.

General comments

- Even *simple* applications might need complex knowledge sources.
- Applications cannot be 100% perfect.
- Applications that are $< 100\%$ perfect can be useful.
- Aids to humans are easier than replacements for humans.

General comments

- Even *simple* applications might need complex knowledge sources.
- Applications cannot be 100% perfect.
- Applications that are $< 100\%$ perfect can be useful.
- Aids to humans are easier than replacements for humans.
- NLP interfaces compete with non-language approaches.

General comments

- Even *simple* applications might need complex knowledge sources.
- Applications cannot be 100% perfect.
- Applications that are $< 100\%$ perfect can be useful.
- Aids to humans are easier than replacements for humans.
- NLP interfaces compete with non-language approaches.
- Typically: shallow processing on arbitrary input or deep processing on narrow domains.

General comments

- Even *simple* applications might need complex knowledge sources.
- Applications cannot be 100% perfect.
- Applications that are $< 100\%$ perfect can be useful.
- Aids to humans are easier than replacements for humans.
- NLP interfaces compete with non-language approaches.
- Typically: shallow processing on arbitrary input or deep processing on narrow domains.
- Limited domain systems require expensive expertise to port or large amounts of (expensive) data.