TDDE16/732A92 Text Mining (2017)

Information extraction

Marco Kuhlmann
Department of Computer and Information Science

LINKÖPING UNIVERSITY

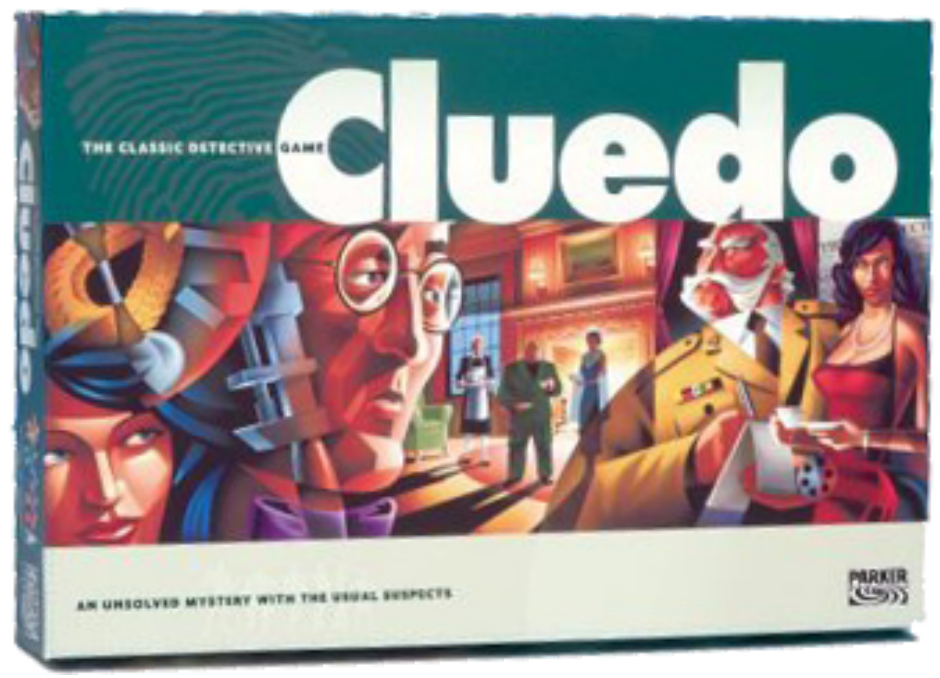
Information extraction

- **Information extraction (IE)** is the task of extracting structured information from running text.

- More specifically, the term ‘structured information’ refers to **named entities** and **semantic relations** between those entities.

  persons, organisations, companies – *X* is-leader-of *Y*, *X* bought *Y*

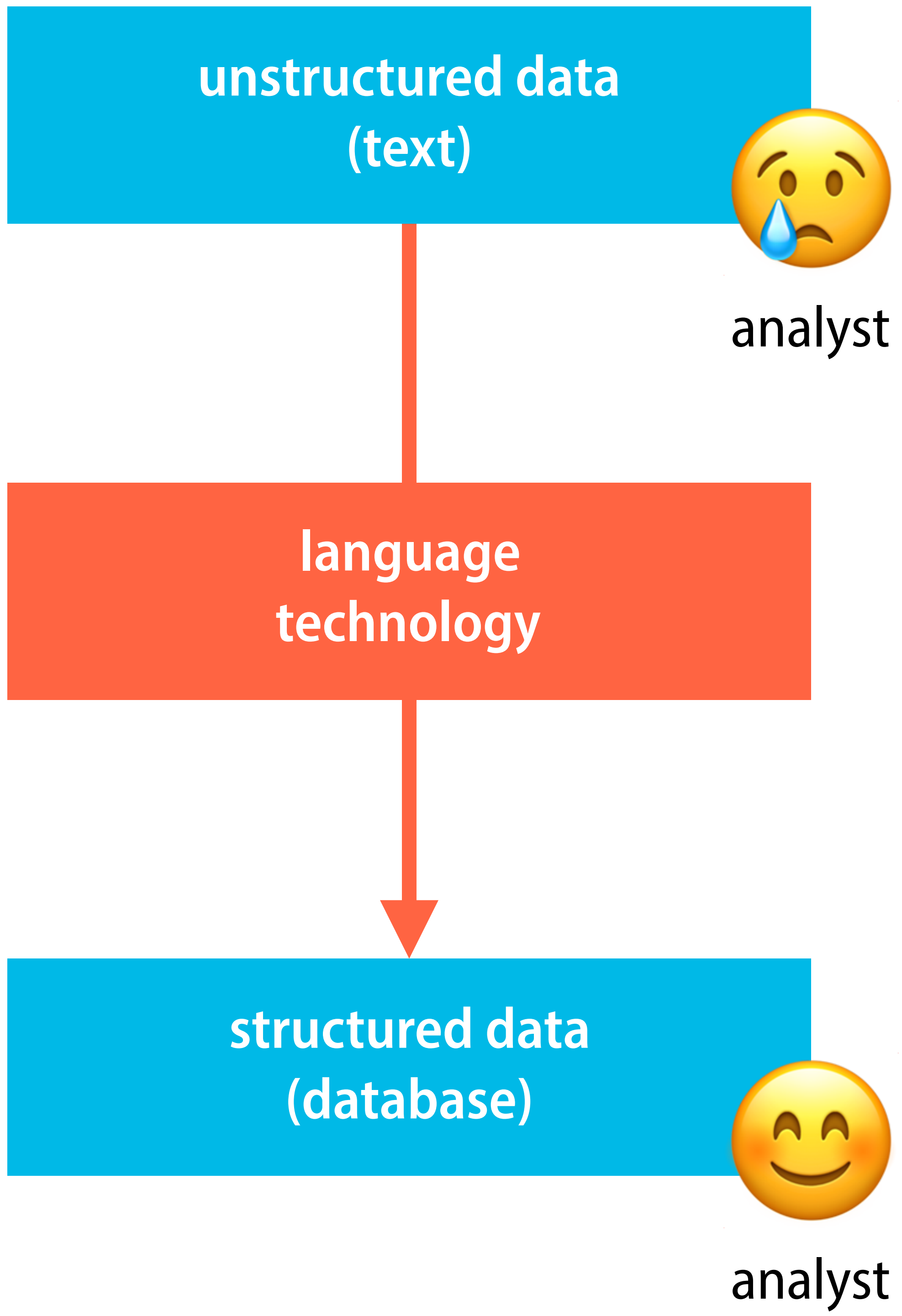
Who did what to whom, where, and when?



named entities

semantic relations

The Knowledge Gap



Information extraction

As of 15 Mar 2002, Hawaii state health officials reported one additional recent case of dengue fever and 6 cases that occurred last year but were not confirmed by laboratory testing until 2002.

Attribute	Value
docno	ProMed.20020322.11
doc_date	2002.03.22
disease_name	dengue fever
norm_stime	2002.03.15
norm_etime	2002.03.15
victim_types	—
location	Hawaii

Source: Grishman et al. (2002)

Why information extraction?

- to find information expressed in natural language

  company acquisitions and mergers

- to create or maintain knowledge bases

  Knowledge Graph, DBPedia

- to support question answering systems

  IBM's Watson

JEOPARDY!

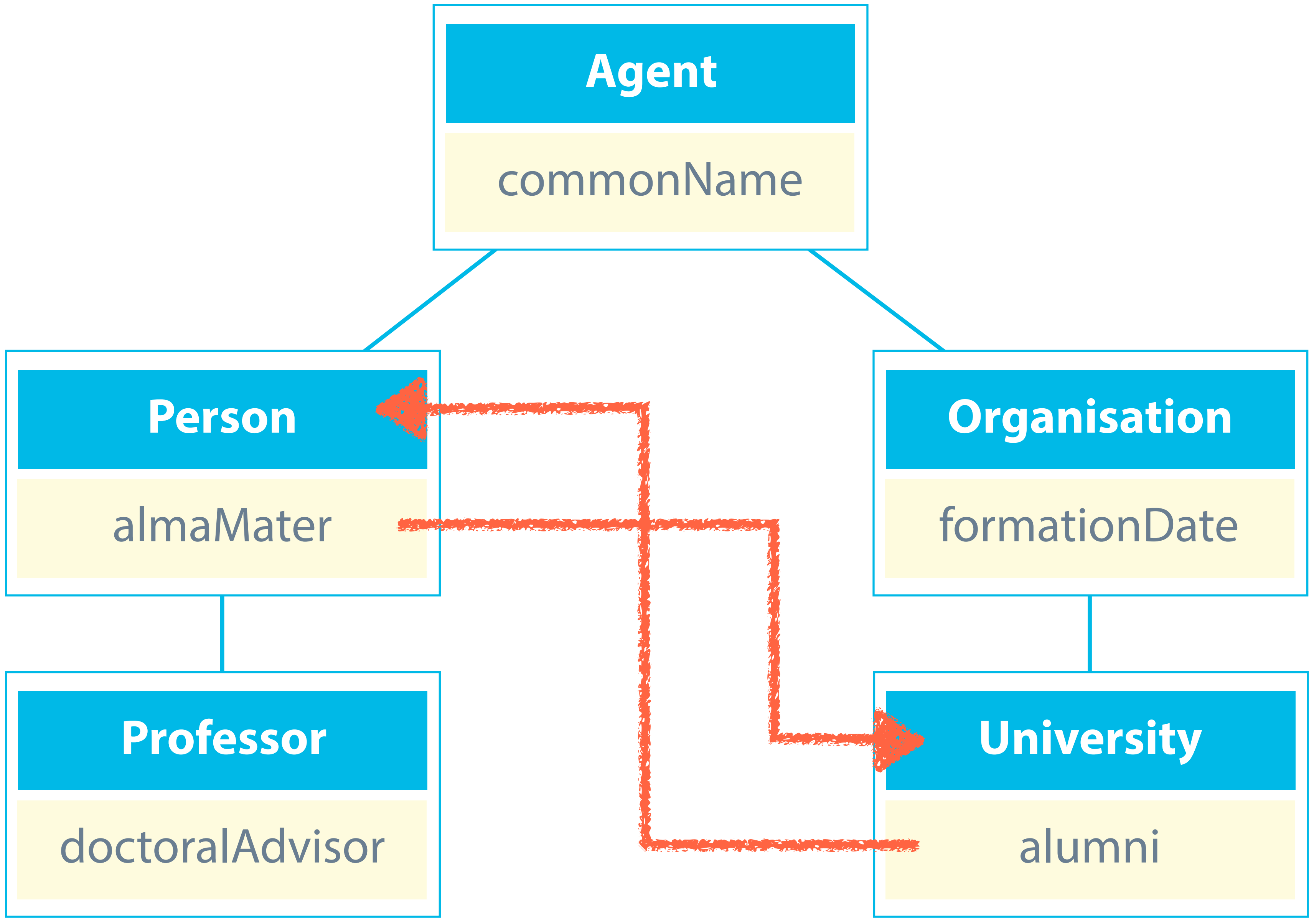
This Stanford University alumnus co-founded educational technology company Coursera.

Source: MacArthur Foundation

SPARQL query against DBPedia

```
SELECT DISTINCT ?x WHERE {
 ?x dbpedia-owl:almaMater dbres:Stanford_University.
 dbres:Coursera dbpedia-owl:founder ?x.
}
```

Part of the DBPedia ontology



http://wiki.dbpedia.org/Ontology

Overview of this lecture

- part-of-speech tagging
- named entity recognition
- dependency parsing
- relation extraction

Part-of-speech tagging

Parts of speech

- A **part of speech** is a category of words that play similar roles within the syntactic structure of a sentence.
- Parts of speech can be defined distributionally or functionally.

  Kim saw the {elephant, movie, mountain, error} before we did.

  verbs = predicates; nouns = arguments; adverbs = modify verbs, …
- There are many different 'tag sets' for parts of speech.

  different languages, different levels of granularity, different design principles

Universal part-of-speech tags

Tag	Category	Examples
ADJ	adjective	*big, old*
ADV	adverb	*very, well*
INTJ	interjection	*ouch!*
NOUN	noun	*girl, cat, tree*
VERB	verb	*run, eat*
PROPN	proper noun	*Mary, John*

Tag	Category	Examples
ADP	adposition	*in, to, during*
AUX	auxiliary verb	*has, was*
CCONJ	conjunction	*and, or, but*
DET	determiner	*a, my, this*
NUM	cardinal numbers	*0, one*
PRON	pronoun	*I, myself, this*

Missing: PART, SCONJ, PUNCT, SYM, X

Source: Universal Dependencies Project

Part-of-speech tagging

- A **part-of-speech tagger** is a computer program that tags each word in a sentence with its part of speech.

- Part-of-speech tagging can be approached as a supervised machine learning problem. This requires training data.

  labelled sentences – words manually tagged with parts-of-speech

- Part-of-speech taggers are commonly evaluated using accuracy, precision, and recall.

Part-of-speech tagging as classification

- Part-of-speech tagging is typically cast as a sequence of classification problems – one classification per word.

- Based on this idea, any method for classification can be used to build a part-of-speech tagger.

  linear classifiers, neural networks

Part-of-speech tagging with a perceptron

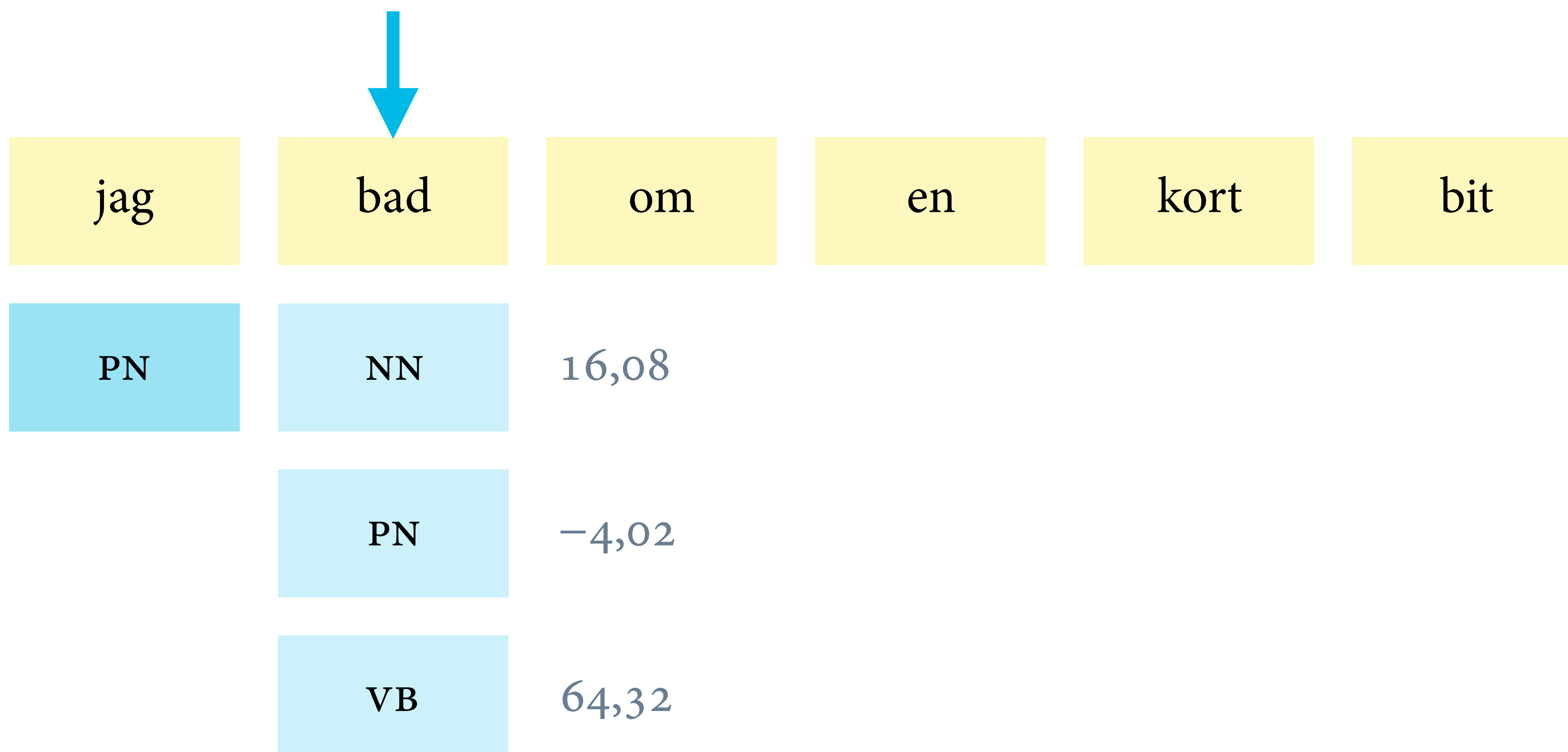
jag	bad	om	en	kort	bit
NN	9,36				
PN	81,72				
VB	−9,18				

Part-of-speech tagging with a perceptron

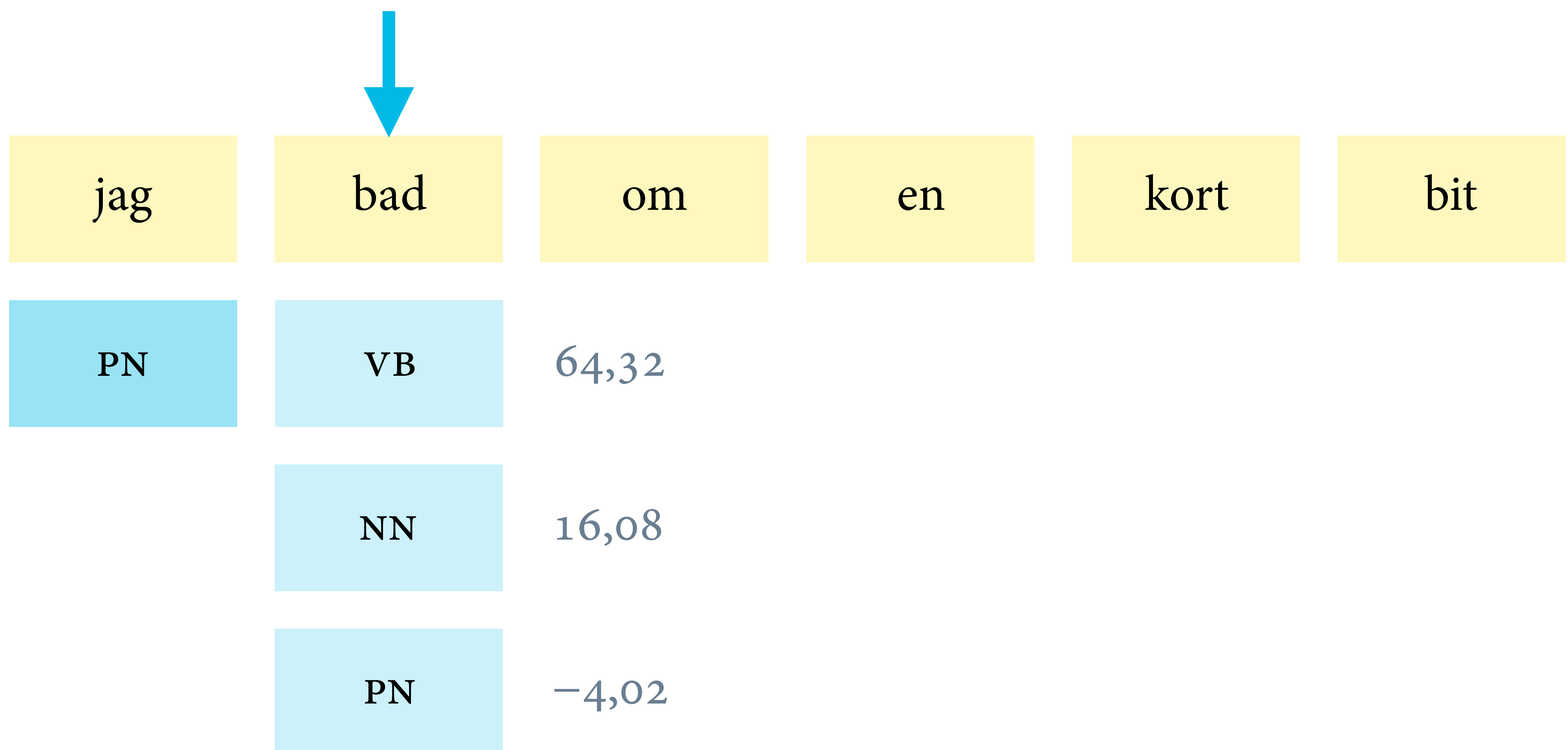
jag bad om en kort bit

PN	81,72
NN	9,36
VB	−9,18

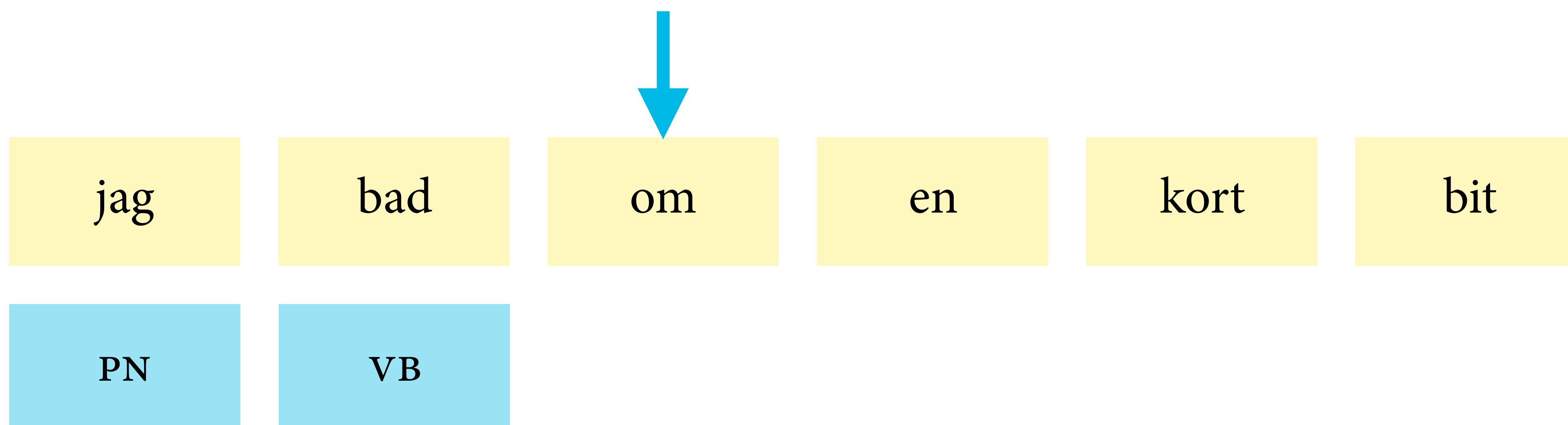
Part-of-speech tagging with a perceptron



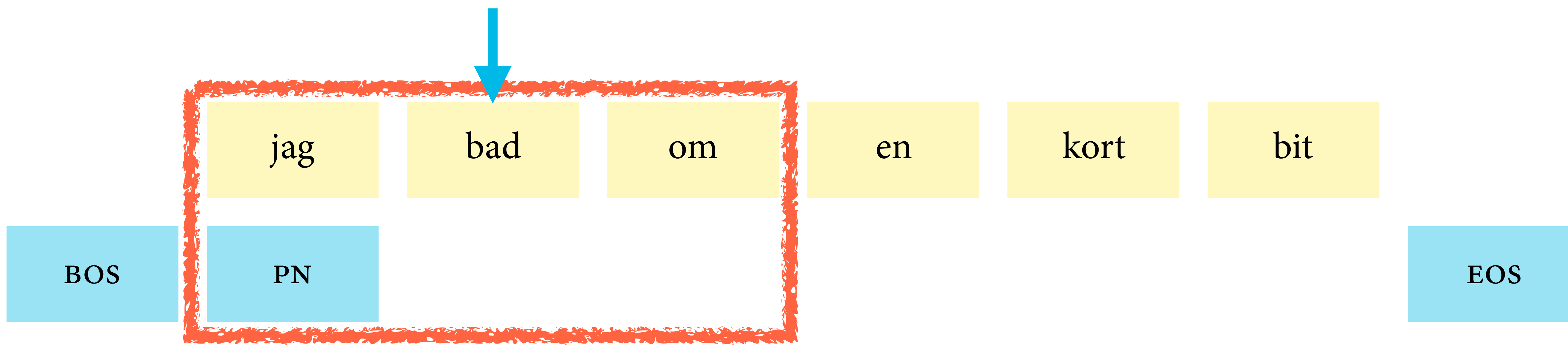
Part-of-speech tagging with a perceptron



Part-of-speech tagging with a perceptron

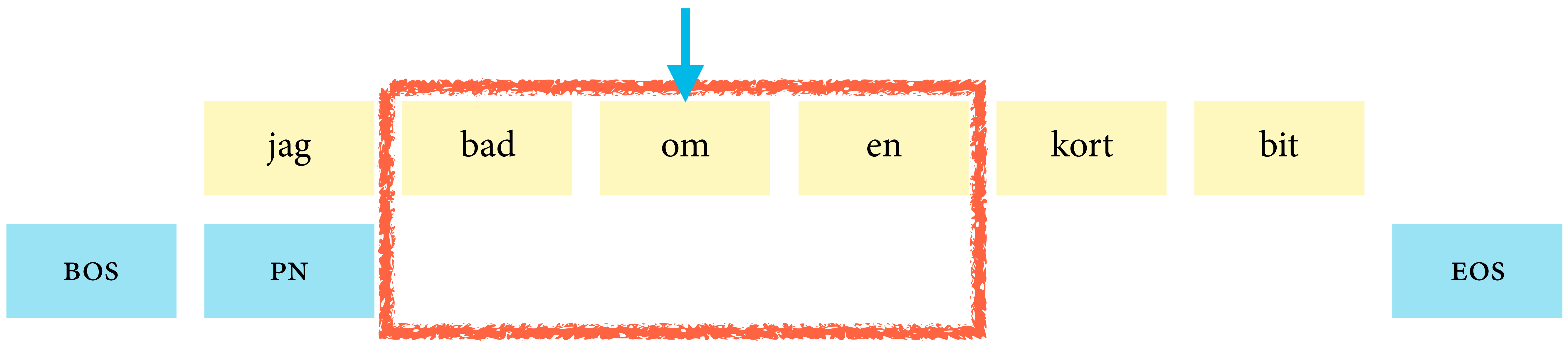


Feature window



With this feature window, we 'see' the current word, the previous word, the next word, and the previous tag.

Feature window



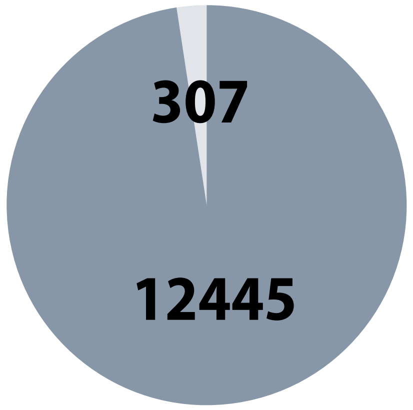
The feature window moves forward during tagging.

Features commonly used in part-of-speech taggers

- identity of current word and neighbouring words
- presence of current word in a tag dictionary
- current word has a particular prefix or suffix
- word shape of current word (USA, IMF → XXX)

Accuracy

	DT	JJ	NN	PP	VB
DT	923	0	0	0	1
JJ	2	1255	132	1	5
NN	0	7	4499	1	18
PP	0	0	0	2332	1
VB	0	5	132	2	3436

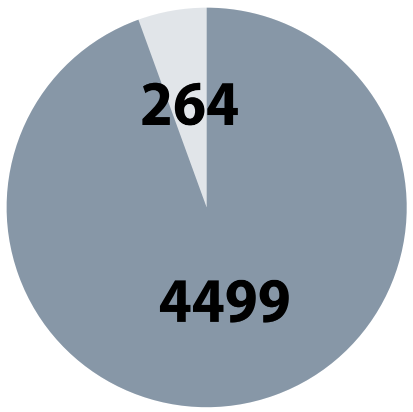


predicted tag

gold-standard tag

Precision with respect to NN

	DT	JJ	NN	PP	VB
DT	923	0	0	0	1
JJ	2	1255	132	1	5
NN	0	7	4499	1	18
PP	0	0	0	2332	1
VB	0	5	132	2	3436

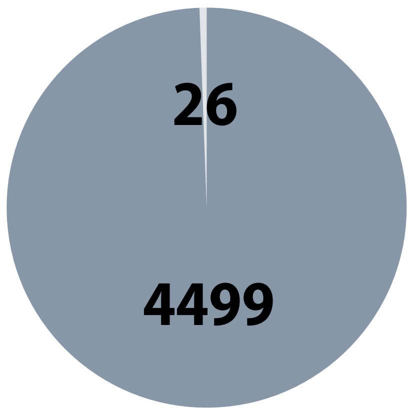


predicted tag

gold-standard tag

Recall with respect to NN

	DT	JJ	NN	PP	VB
DT	923	0	0	0	1
JJ	2	1255	132	1	5
NN	0	7	4499	1	18
PP	0	0	0	2332	1
VB	0	5	132	2	3436



predicted tag

gold-standard tag

Named entity recognition

Named entity recognition

Named entity recognition (NER) is the task of finding mentions of named entities in a text, and labelling them with their type.

person, company, organisation, geopolitical entity

Named entities, example

Citing high fuel prices, [ORG United Airlines] said [TIME Friday] it has increased fares by [MONEY $6] per round trip on flights to some cities also served by lower-cost carriers. [ORG American Airlines], a unit of [ORG AMR Corp.], immediately matched the move, spokesman [PER Tim Wagner] said. [ORG United], a unit of [ORG UAL Corp.], said the increase took effect [TIME Thursday] and applies to most routes where it competes against discount carriers, such as [LOC Chicago] to [LOC Dallas] and [LOC Denver] to [LOC San Francisco].

Example from Jurafsky and Martin (2017)

Properties of named entities

- can be referred to with a proper name
- can be indexed and linked to
- participate in semantic relations
- are common answers in question answering systems
- can be associated with attitudes

Types of named entities in DBPedia

- **Persons:** Actor, Curler, FictionalCharacter
- **Organisations:** Band, Company, SportsTeam
- **Places:** Building, Mountain, Country
- **Dates and times:** Date, Year, HistoricalPeriod
- **Medical terms:** Muscle, Enzyme, Disease

Inflected names in Polish

Case	Form
Nominative	Muammar Kaddafi
Genitive	Muammara Kaddafiego
Dative	Muammarowi Kaddafiemu
Accusative	Muammara Kaddafiego
Instrumental	Muammarem Kaddafim
Locative	Muammarze Kaddafim
Vocative	Muammarze Kaddafi

Piskorsky and Yangarber (2013)

Type ambiguities

- [PER Washington] was born into slavery.
- [ORG Washington] went up 2 games to 1 in the four-game series.
- Blair arrived in [LOC Washington] for his last state visit.
- In June, [GPE Washington] passed a primary seatbelt law.
- The [VEH Washington] had proved to be a leaky ship,…

Examples adapted from Jurafsky and Martin (2017)

Named entity recognition as sequence labelling

- State-of-the algorithms treat named entity recognition as a word-by-word sequence labelling task.

  Just as part-of-speech tagging!

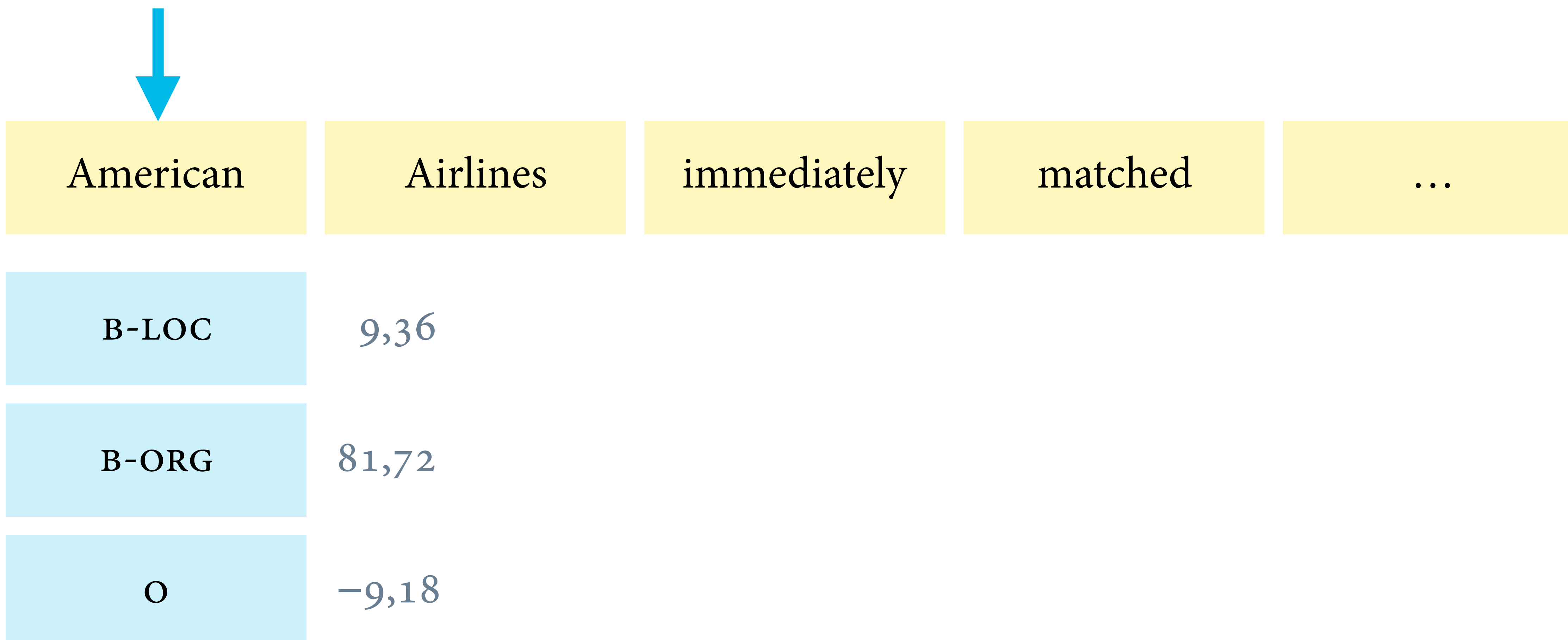
- The basic idea is to use tags that can encode both the boundaries and the types of named entity mentions.

- A common encoding is the **IOB scheme**, where there is a tag for the beginning (B) and inside of each entity type, as well as an additional tag for tokens outside (O) any entity.

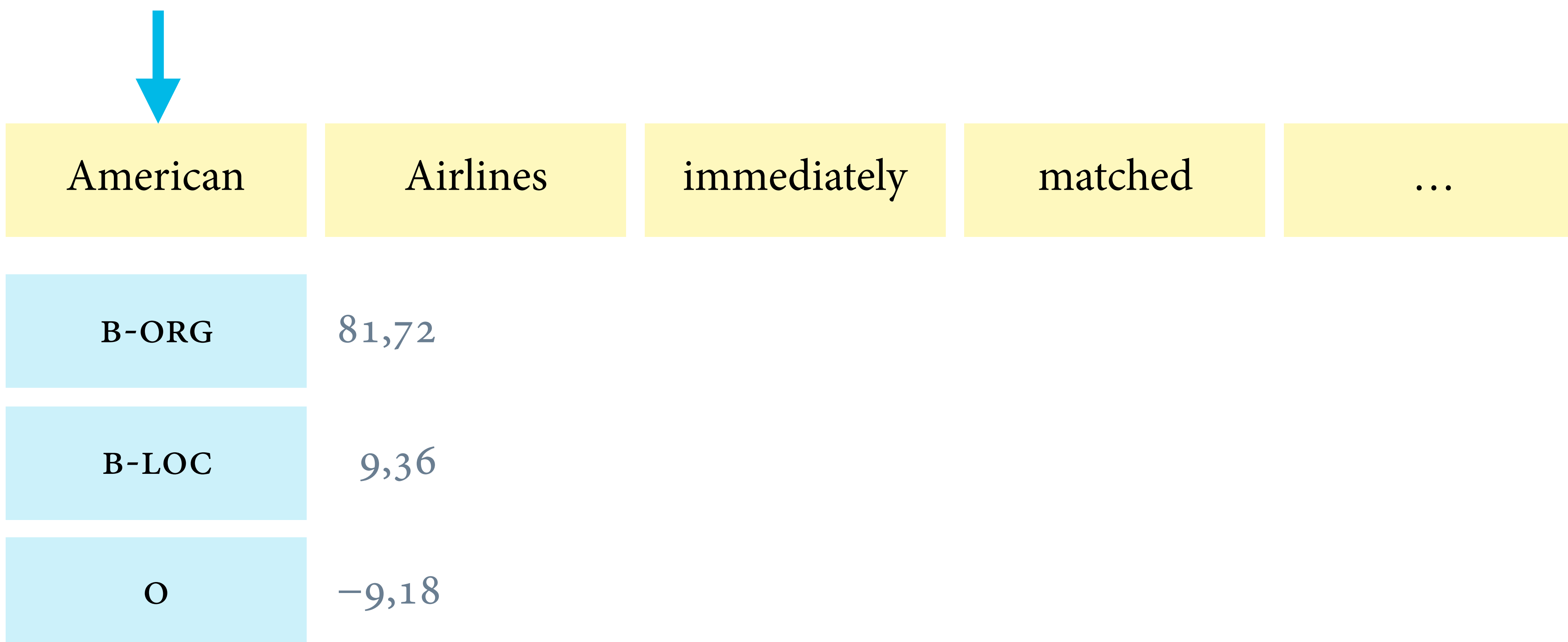
Named entity recognition as sequence labelling

Token	IOB tag
American	B-ORG
Airlines	I-ORG
immediately	O
matched	O
the	O
move	O
Wagner	B-PER
said	O
.	O

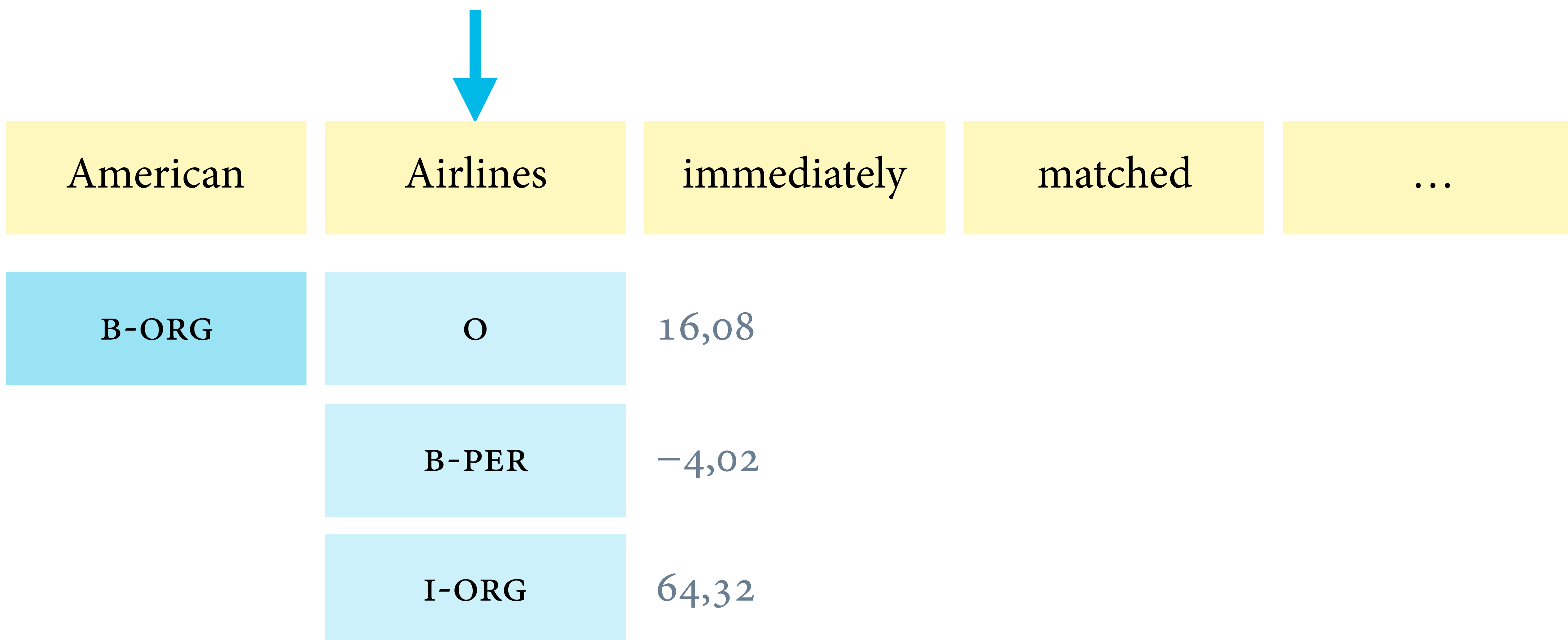
Named entity tagging with a sequence classifier



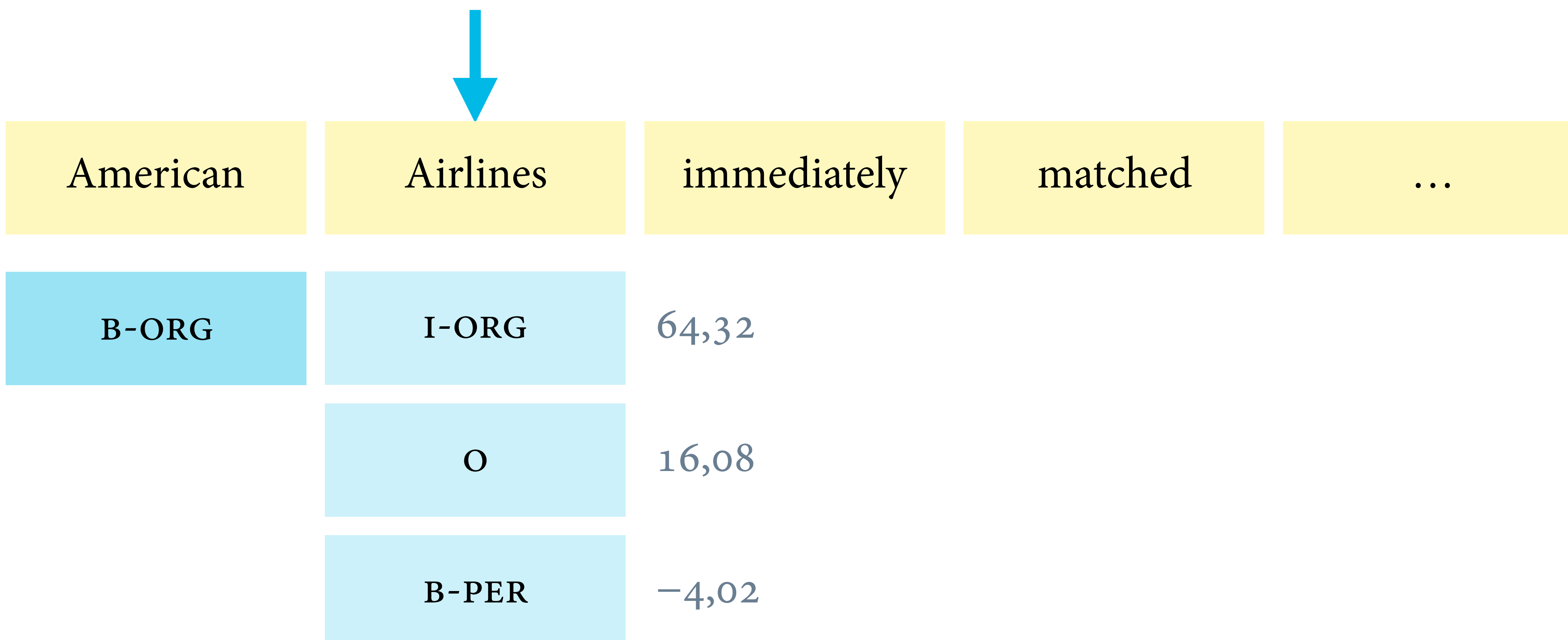
Named entity tagging with a sequence classifier



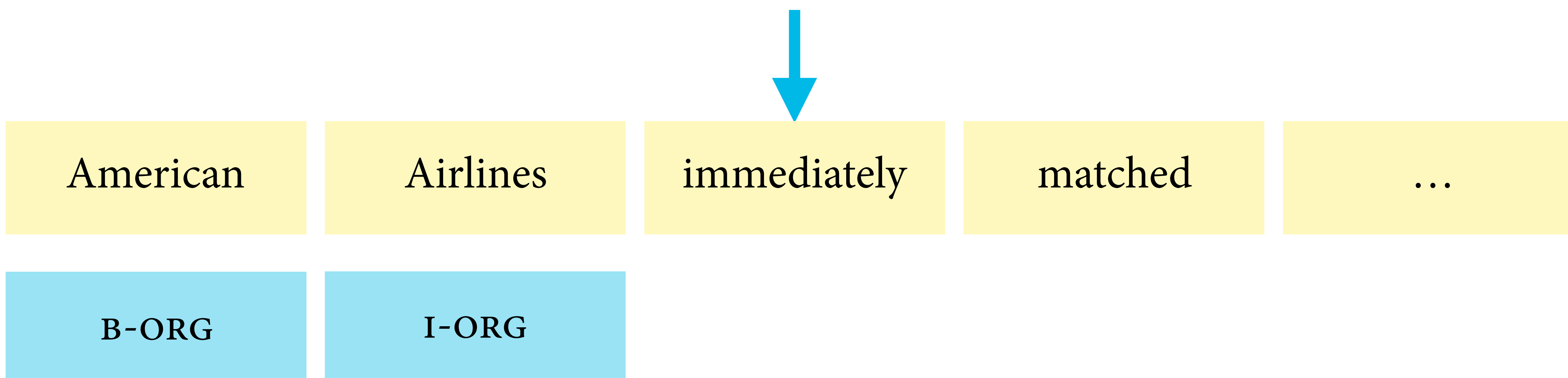
Named entity tagging with a sequence classifier



Named entity tagging with a sequence classifier



Named entity tagging with a sequence classifier



Features commonly used in NER systems

- identity of current word and neighbouring words
- part-of-speech of current word and neighbouring words
- presence of current word in a gazetteer
- current word has a particular prefix or suffix
- word shape of current word (USA, IMF → XXX)
- syntactic contexts (dependency trees)

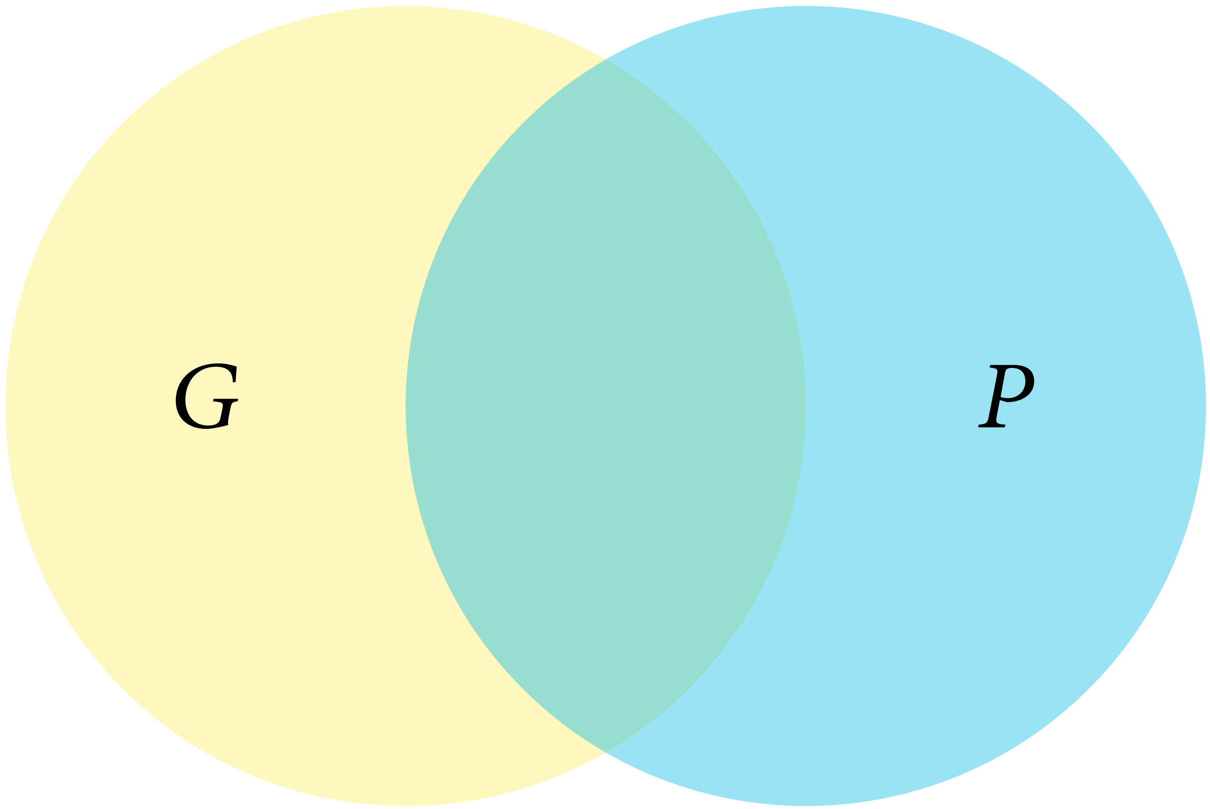
Adapted from Jurafsky and Martin (2017)

Example of a gazetteer

Ale Alingsås Alvesta Aneby Arboga Arjeplogs Arvidsjaurs Arvika Askersunds Avesta Bengtsfors Bergs Bjurholms Bjuvs Bodens Bollebygds Bollnäs Borgholms Borlänge Borås Botkyrka Boxholms Bromölla Bräcke Burlövs Båstads Dals-Eds Danderyds Degerfors Dorotea Eda Ekerö Eksjö Emmaboda Enköpings Eskilstuna Eslövs Essunga Fagersta Falkenbergs Falköpings Falu Filipstads Finspångs Flens Forshaga Färgelanda Gagnefs Gislaveds Gnesta Gnosjö Gotlands Grums Grästorps Gullspångs Gällivare Gävle Göteborgs Götene Habo Hagfors Hallsbergs Hallstahammars Halmstads Hammarö Haninge Haparanda Heby Hedemora Helsingborgs Herrljunga Hjo Hofors Huddinge Hudiksvalls Hultsfreds Hylte Håbo Hällefors Härjedalens Härnösands Härryda Hässleholms Höganäs Högsby Hörby Höörs Jokkmokks Järfälla Jönköpings Kalix Kalmar Karlsborgs Karlshamns Karlskoga Karlskrona Karlstads Katrineholms Kils Kinda Kiruna Klippans Knivsta Kramfors Kristianstads Kristinehamns Krokoms Kumla Kungsbacka Kungsörs Kungälvs Kävlinge Köpings Laholms Landskrona Laxå Lekebergs Leksands Lerums Lessebo Lidingö Lidköpings Lilla Edets Lindesbergs Linköpings Ljungby Ljusdals Ljusnarsbergs Lomma Ludvika Luleå Lunds Lycksele Lysekils Malmö Malung-Sälens Malå Mariestads Marks Markaryds Melleruds Mjölby Mora Motala Mullsjö Munkedals Munkfors Mölndals Mönsterås Mörbylånga Nacka Nora Norbergs Nordanstigs Nordmalings Norrköpings Norrtälje Norsjö Nybro Nykvarns Nyköpings Nynäshamns Nässjö Ockelbo Olofströms Orsa Orusts Osby Oskarshamns Ovanåkers Oxelösunds Pajala Partille Perstorps Piteå Ragunda Robertsfors Ronneby Rättviks Sala Salems Sandvikens Sigtuna Simrishamns Sjöbo Skara Skellefteå Skinnskattebergs Skurups Skövde Smedjebackens Sollefteå Sollentuna Solna Sorsele Sotenäs Staffanstorps Stenungsunds Stockholms Storfors Storumans Strängnäs Strömstads Strömsunds Sundbybergs Sundsvalls Sunne Surahammars Svalövs Svedala Svenljunga Säffle Säters Sävsjö Söderhamns Söderköpings Södertälje Sölvesborgs Tanums Tibro Tidaholms Tierps Timrå Tingsryds Tjörns Tomelilla Torsby Torsås Tranemo Tranås Trelleborgs Trollhättans Trosa Tyresö Täby Töreboda Uddevalla Ulricehamns Umeå Upplands Väsby Upplands-Bro Uppsala Uppvidinge Vadstena Vaggeryds Valdemarsviks Vallentuna Vansbro Vara Varbergs Vaxholms Vellinge Vetlanda Vilhelmina Vimmerby Vindelns Vingåkers Vårgårda Vänersborgs Vännäs Värmdö Värnamo Västerviks Västerås Växjö Ydre Ystads Åmåls Ånge Åre Årjängs Åsele Åstorps Åtvidabergs Älmhults Älvdalens Älvkarleby Älvsbyns Ängelholms Öckerö Ödeshögs Örebro Örkelljunga Örnsköldsviks Östersunds Österåkers Östhammars Östra Göinge Överkalix Övertorneå

Evaluation of named entity recognition

gold-standard entities



predicted entities

$$\text{precision} = \frac{|G \cap P|}{|P|} \qquad \text{recall} = \frac{|G \cap P|}{|G|}$$

Issues in the evaluation of NER systems

		gold standard	system
1	First	B–ORG	O
2	Bank	I–ORG	B–ORG
3	of	I–ORG	I–ORG
4	Chicago	I–ORG	I–ORG
5	announced	O	O
	…		

ORG 1–4

ORG 2–4

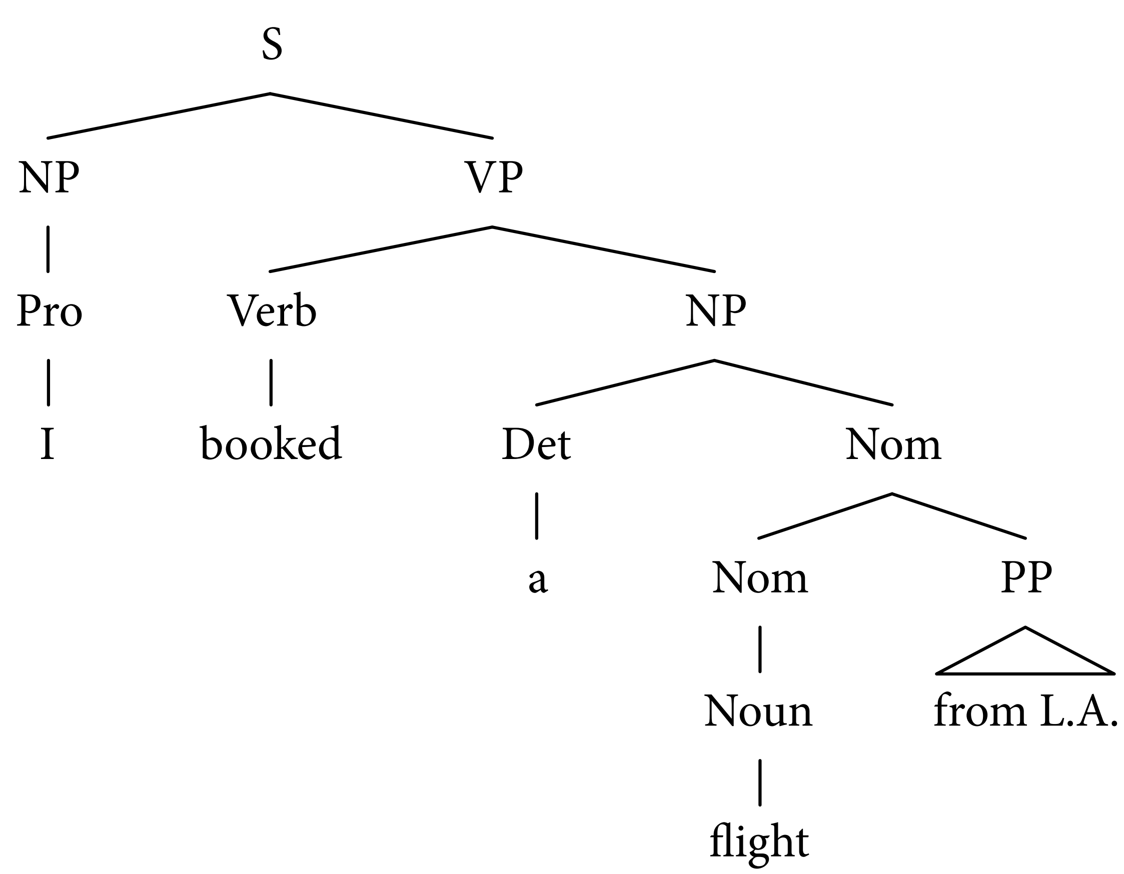
Dependency parsing

Dependency parsing

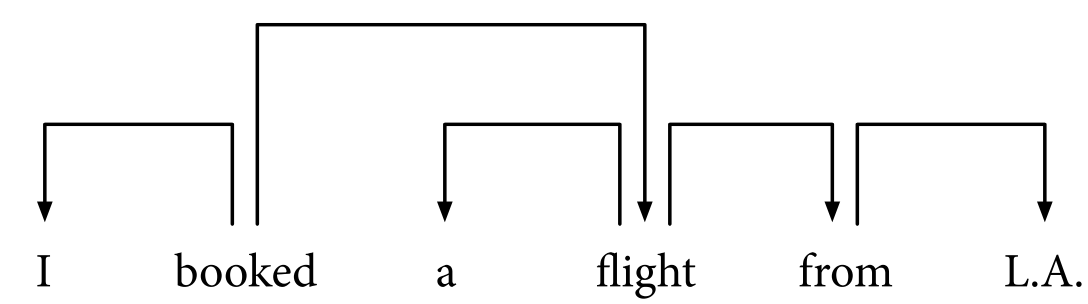
Dependency parsing is the task of mapping a natural language sentence into a formal representation of its syntactic structure in the form of a dependency tree.

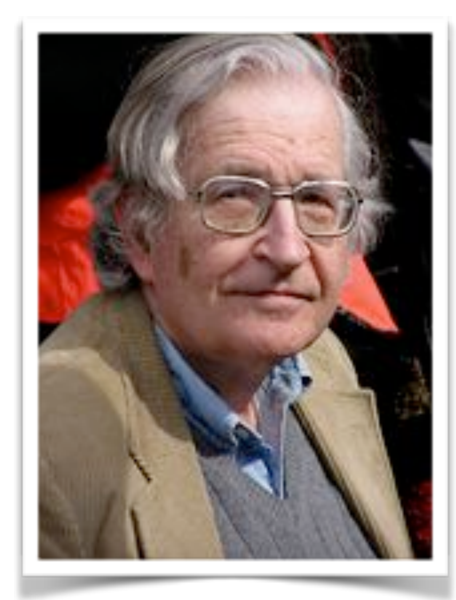
Different syntactic representations

Phrase structure tree

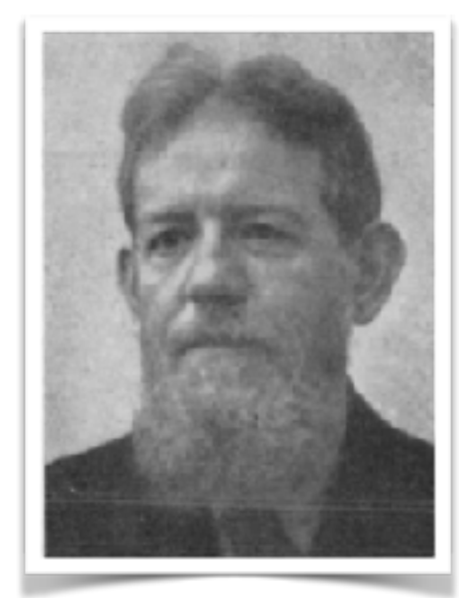


Dependency tree





Noam Chomsky



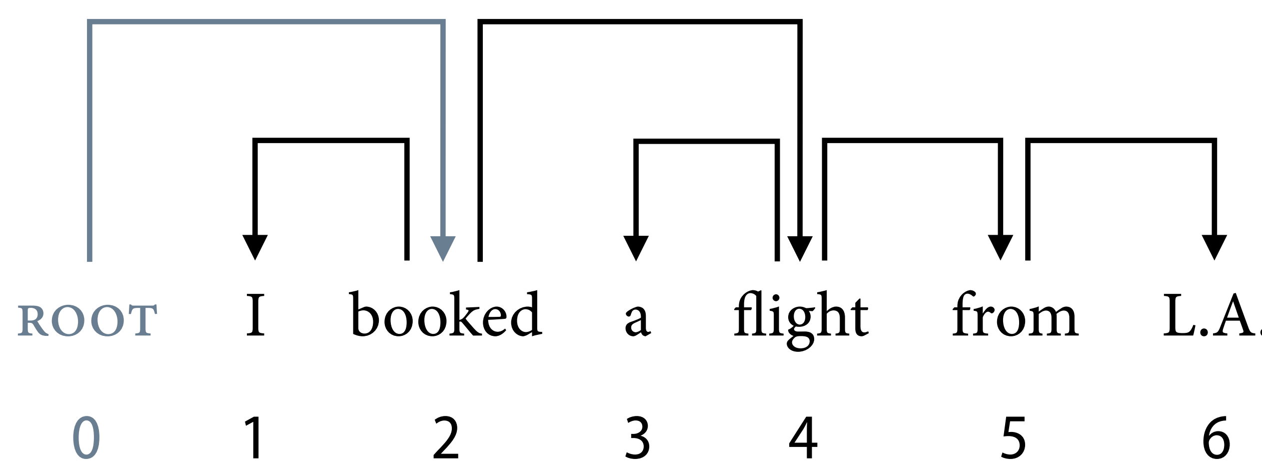
Lucien Tesnière

Source: Wikimedia Commons [1] [2]

Dependency trees

A **dependency tree** for an n-word sentence x is a directed graph $G = (V, A)$ where $V = \{1, \ldots, n\}$ and where there exists a vertex $r \in V$ such that every vertex $v \in V$ is reachable from r via exactly one directed path. The vertex $r \in V$ is called the **root** of the G.

Representation of dependency trees



0	1	2	3	4	5	6
0	2	0	4	2	4	5

The tree is represented by the list of its head positions.

Dependency parsing as classification

- We have seen how part-of-speech tagging can be broken down into a sequence of classification problems.
- The same idea can be applied to dependency parsing.
- Instead of POS tags, the classifier will predict **transitions** that take the parser from one **configuration** to another.

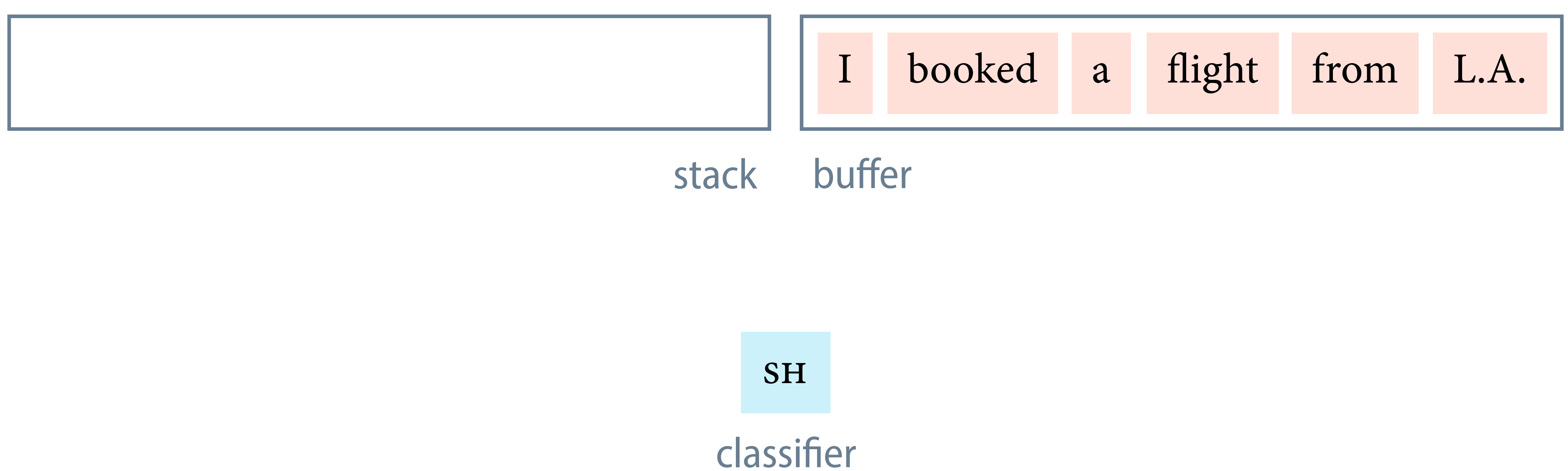
  moves, states

Transition-based dependency parsing

- The parser starts in the **initial configuration**.
- It then calls the classifier, which predicts the transition that the parser should make to move to the next configuration.
- This process is repeated until the parser reaches a **terminal configuration**.

Transition-based dependency parsing, example

I booked a flight from L.A.



Transition-based dependency parsing, example

I booked a flight from L.A.

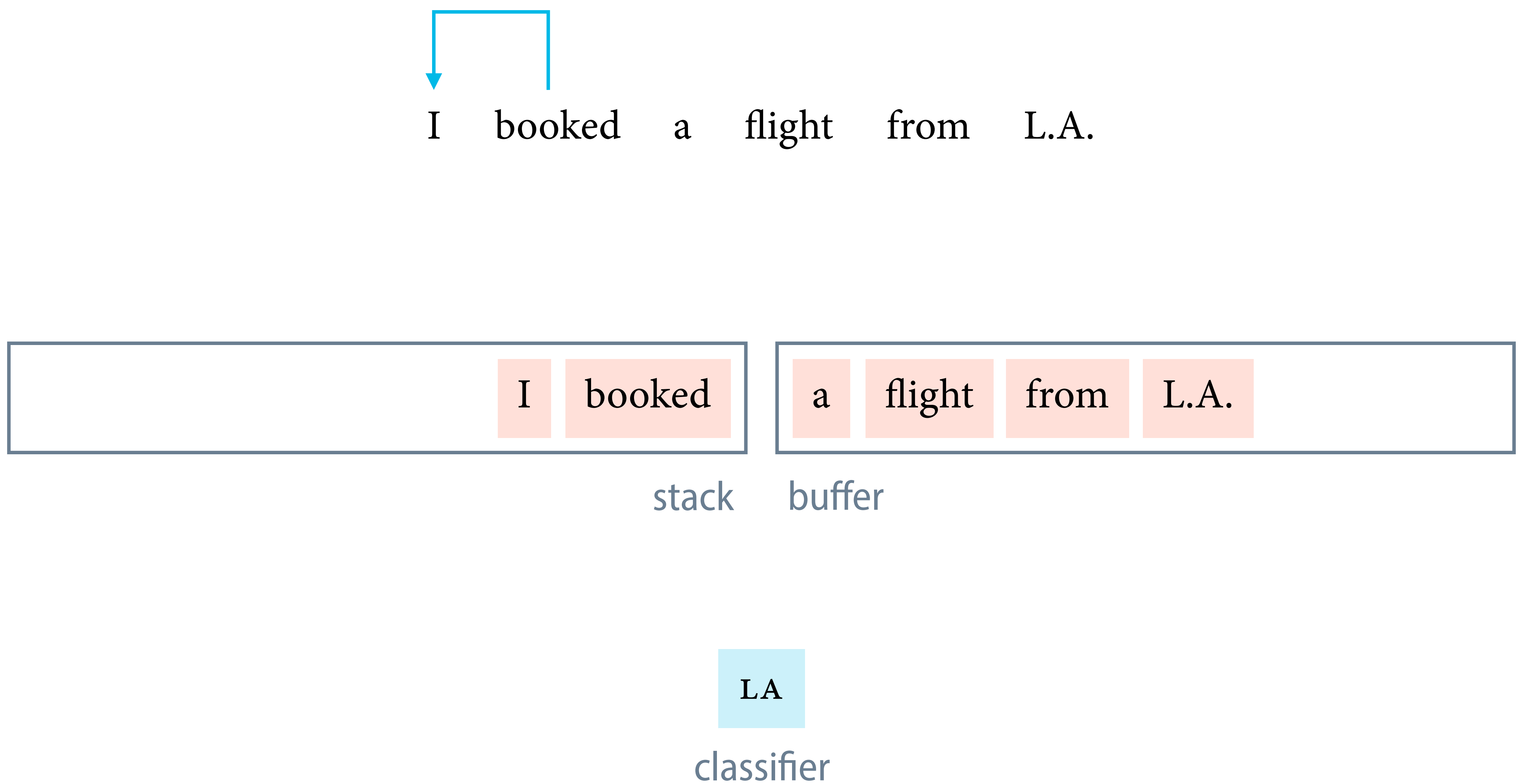
I | booked a flight from L.A.

stack | buffer

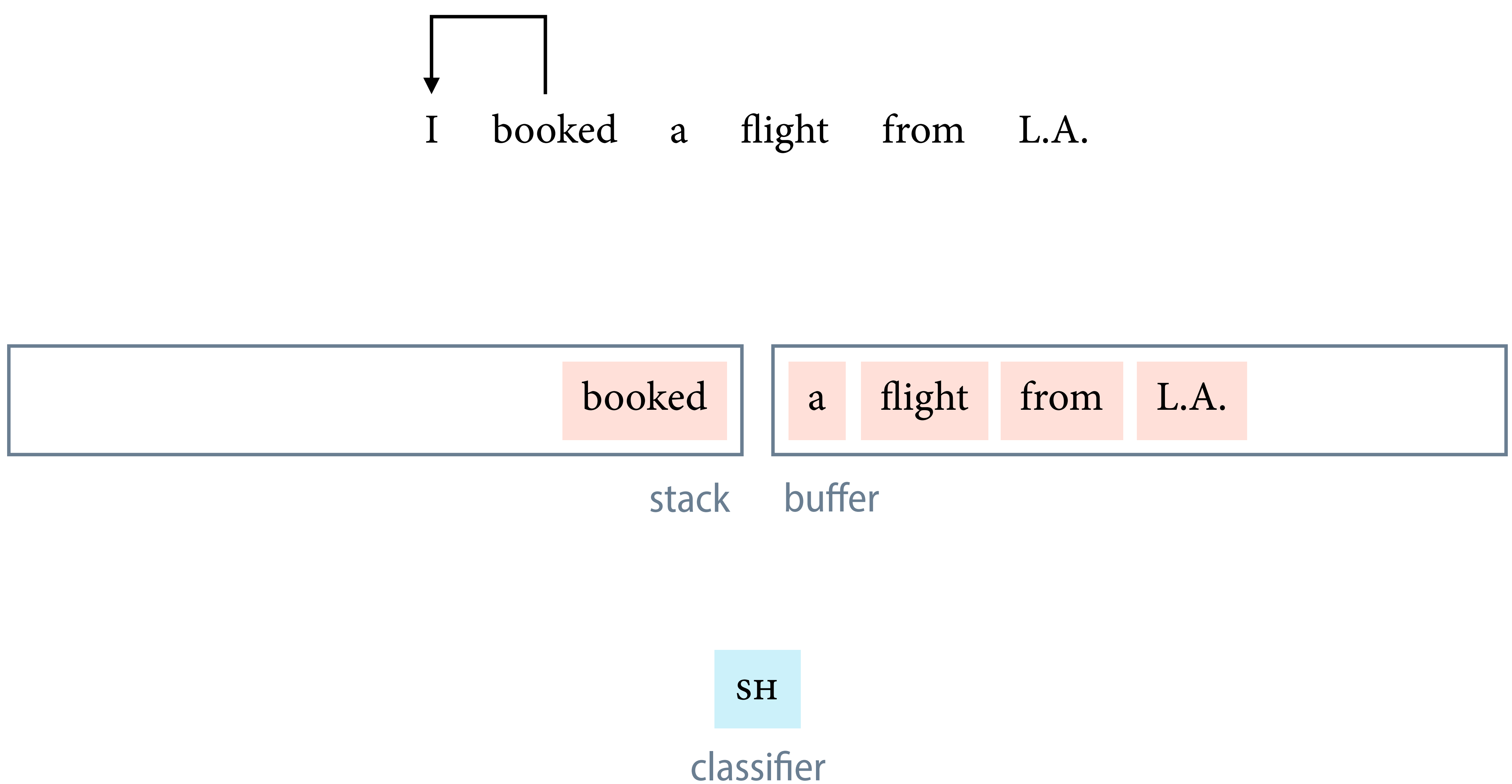
SH

classifier

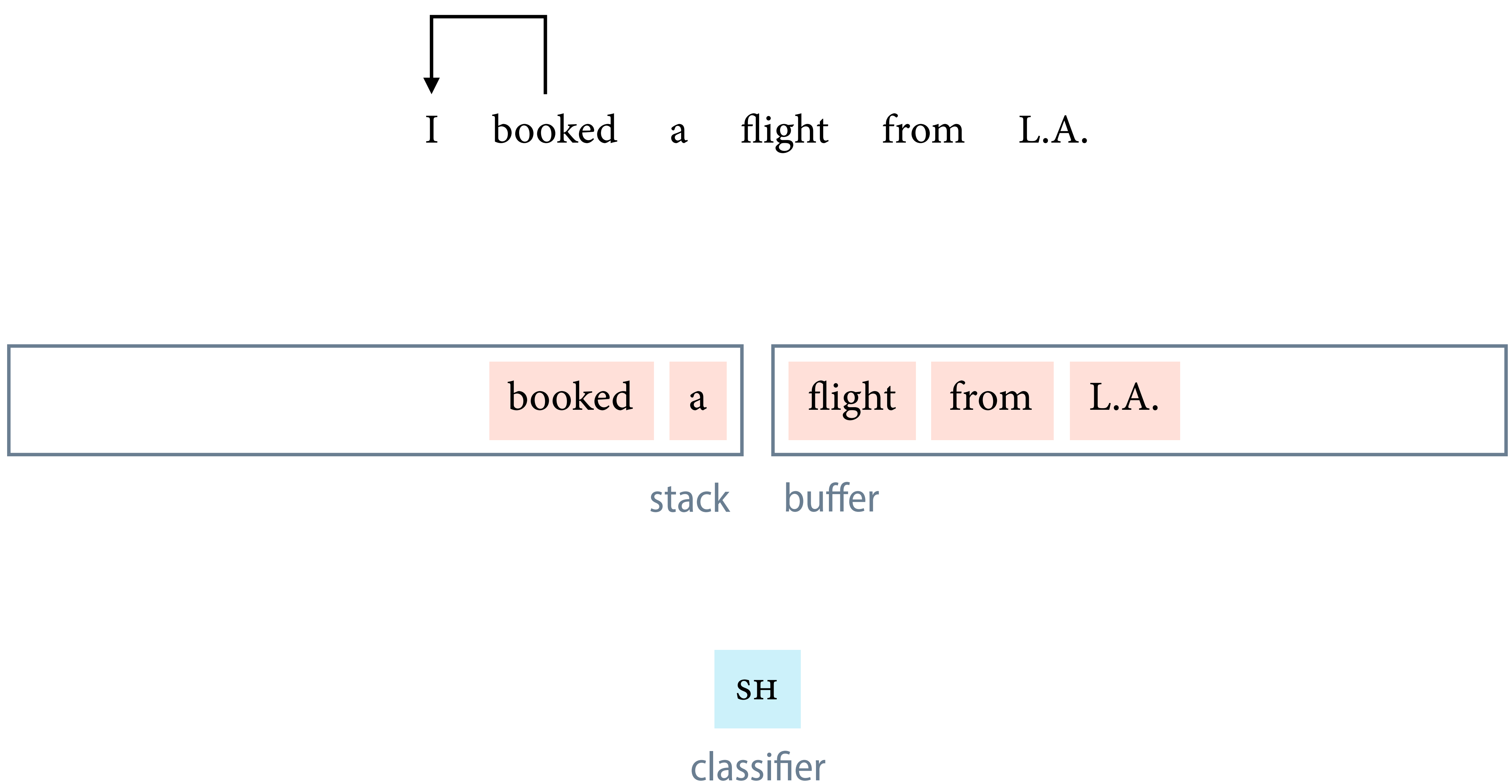
Transition-based dependency parsing, example



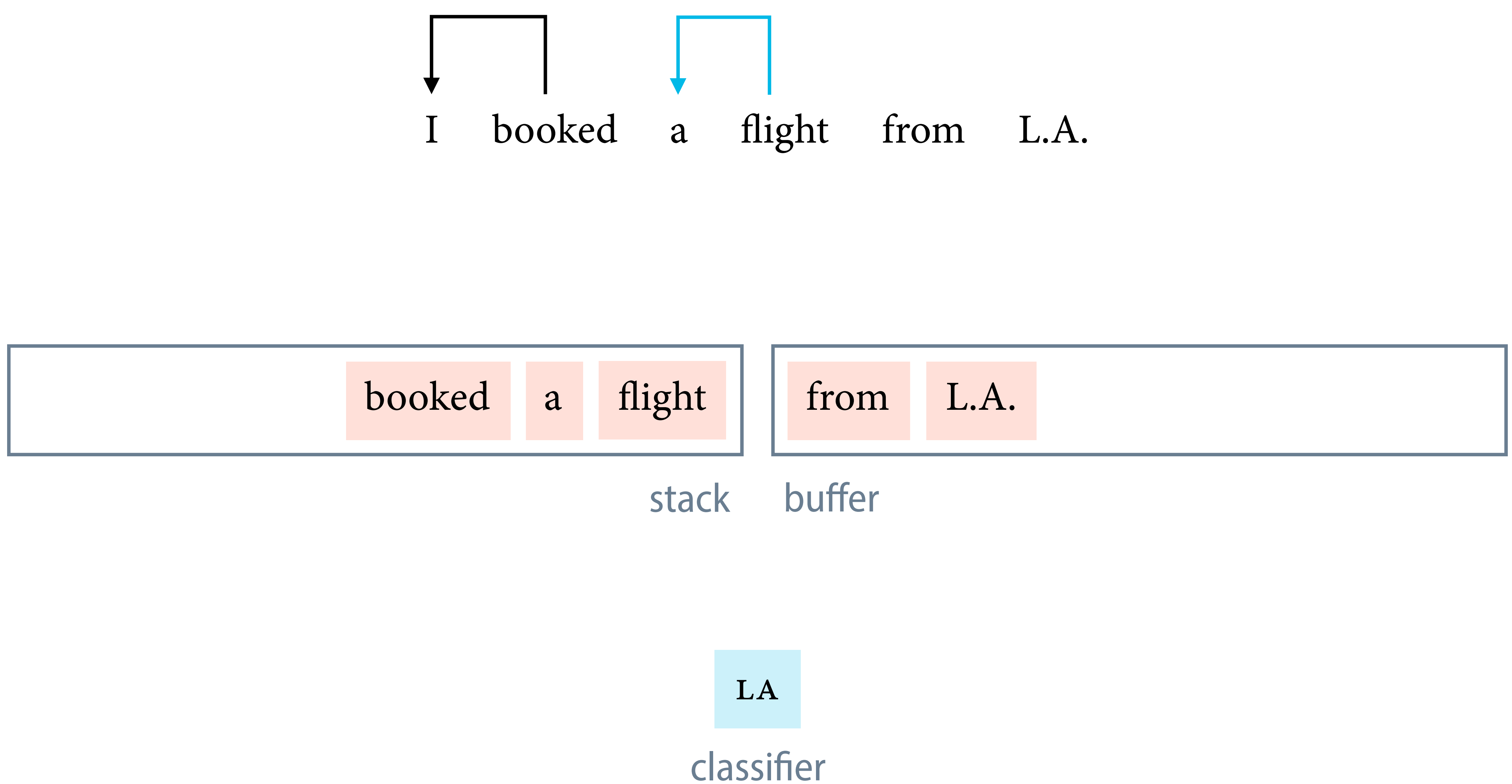
Transition-based dependency parsing, example



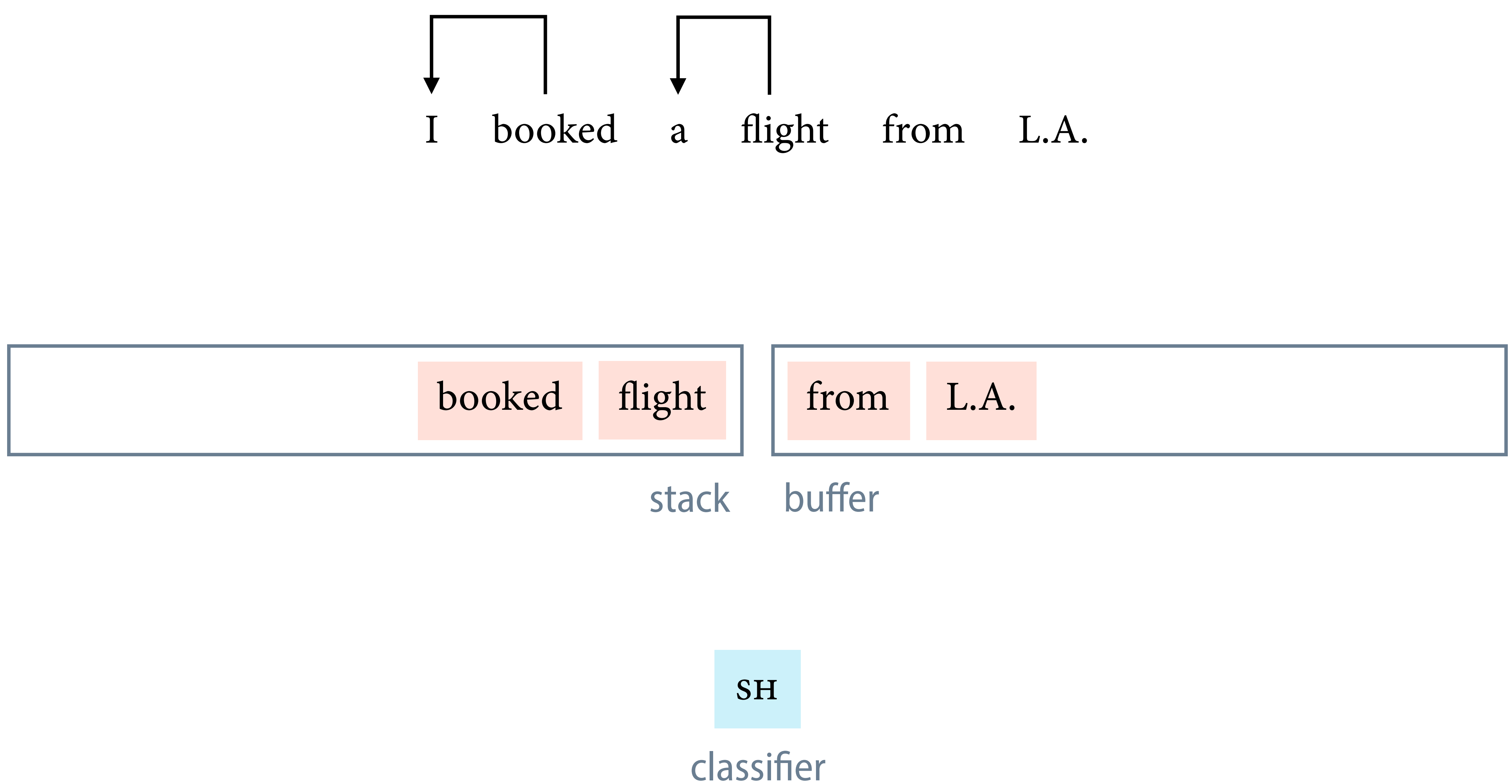
Transition-based dependency parsing, example



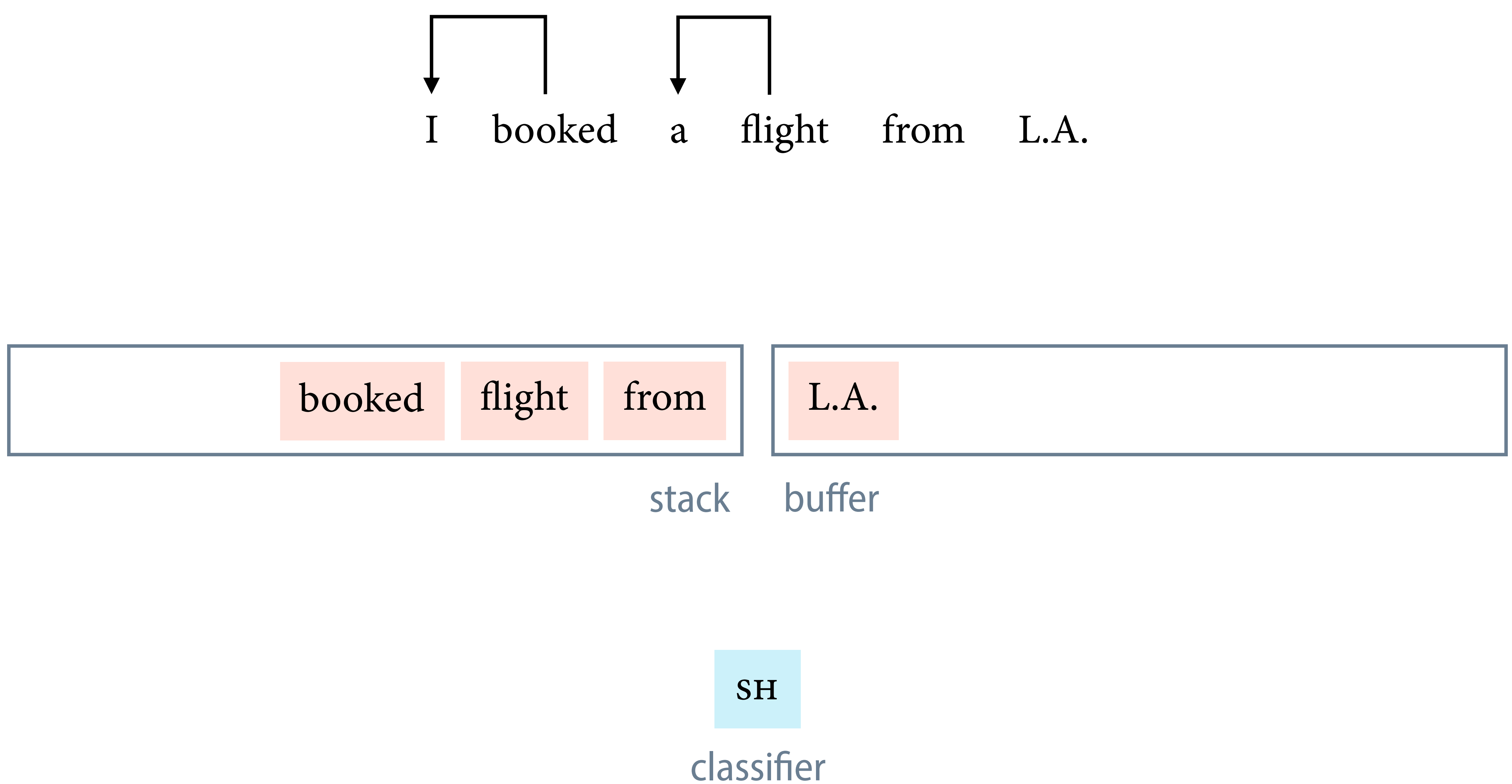
Transition-based dependency parsing, example



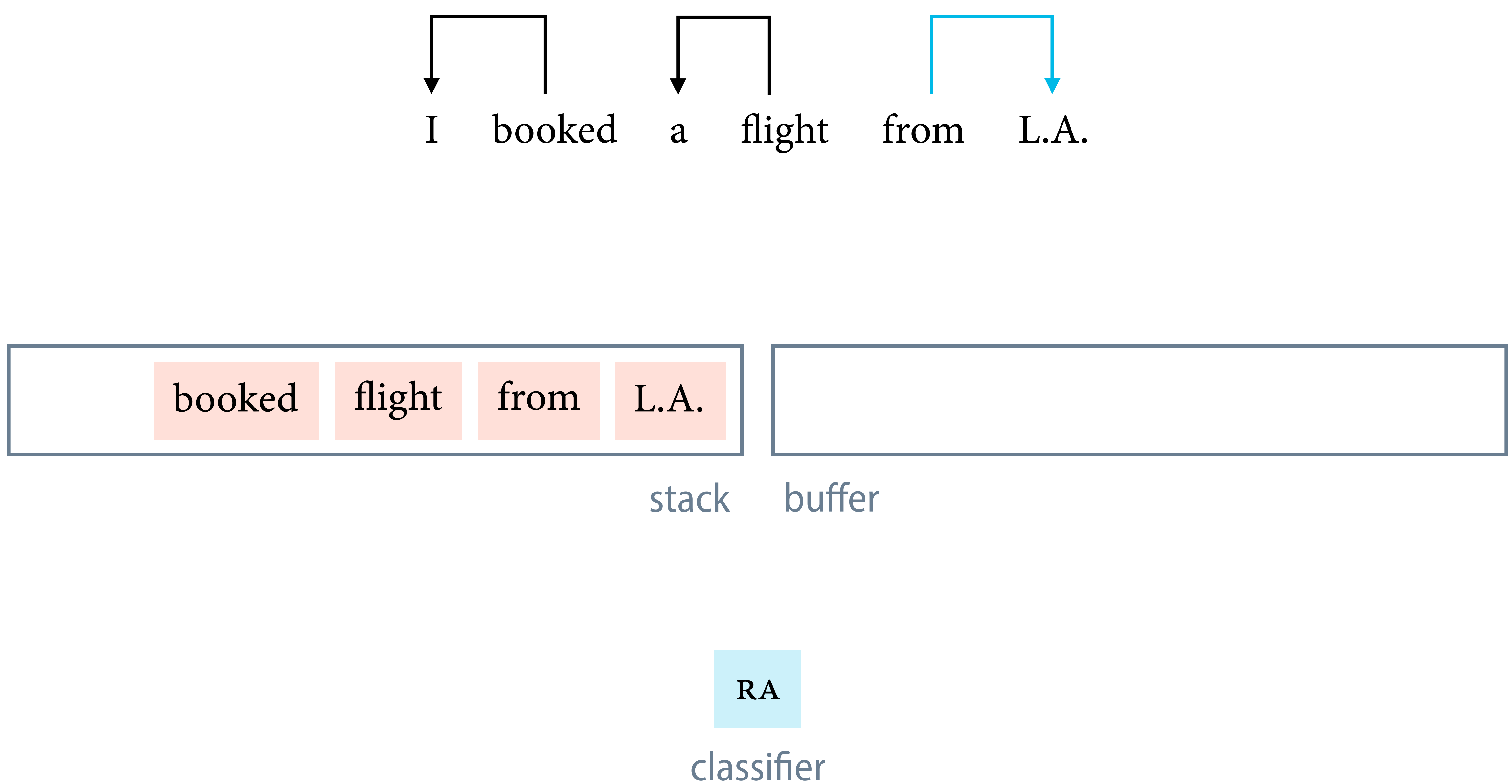
Transition-based dependency parsing, example



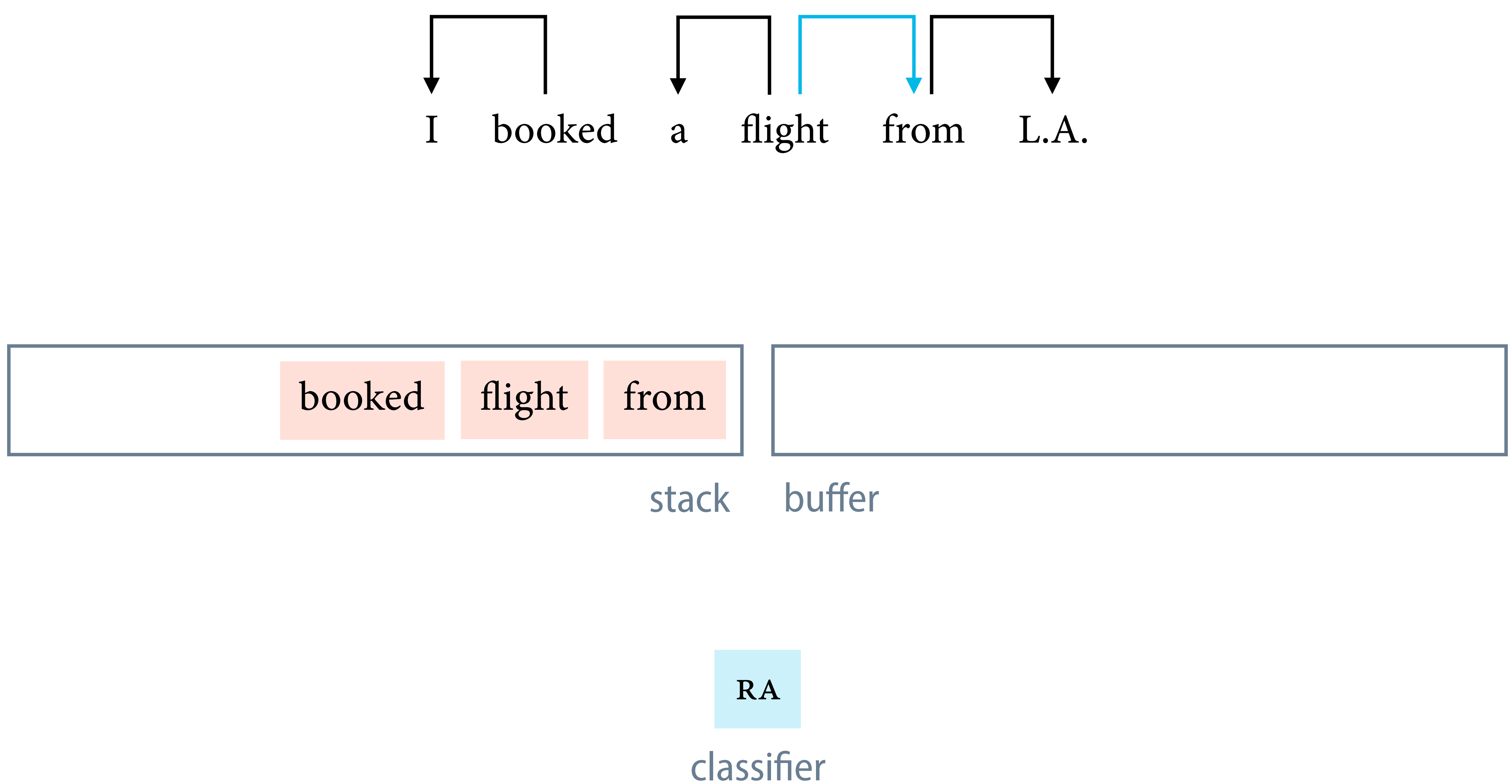
Transition-based dependency parsing, example



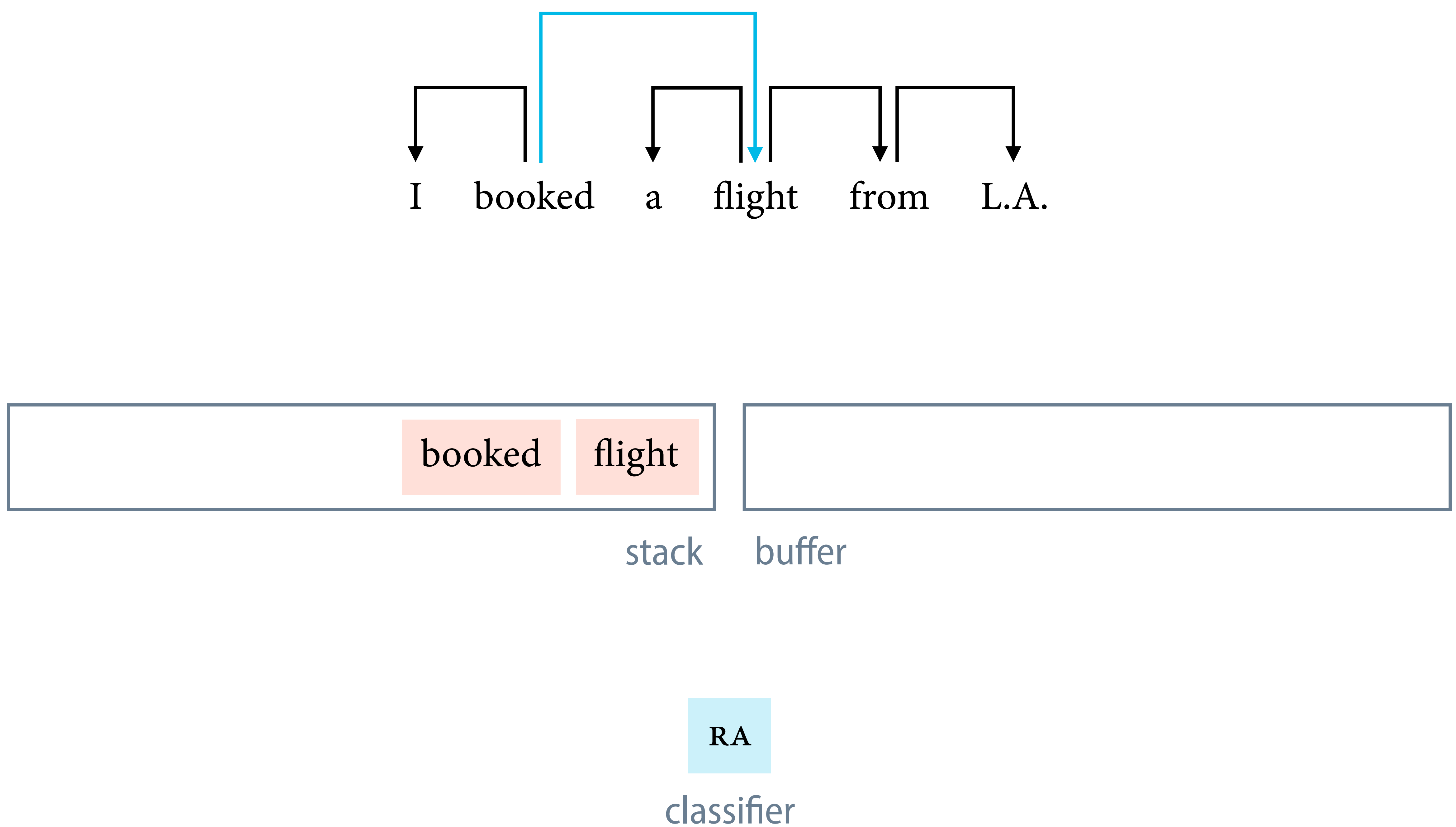
Transition-based dependency parsing, example



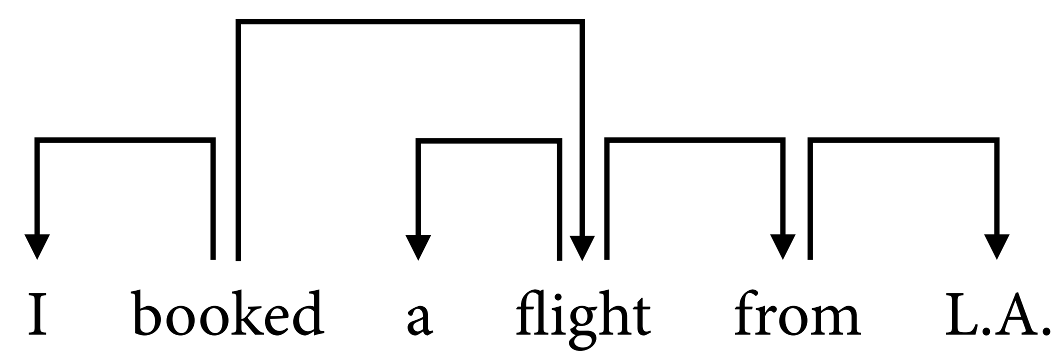
Transition-based dependency parsing, example

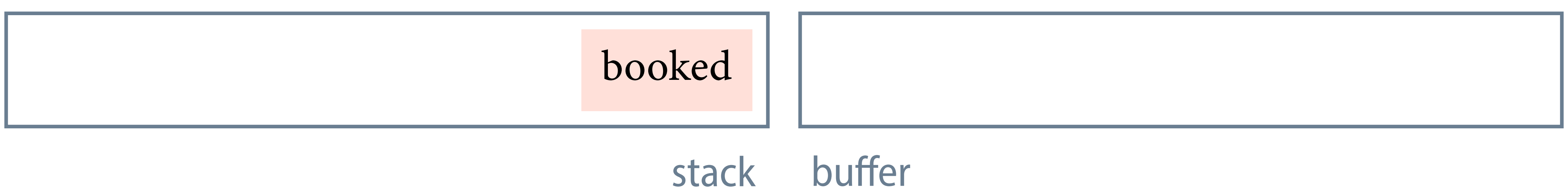


Transition-based dependency parsing, example



Transition-based dependency parsing, example





(terminal configuration)

Features in transition-based dependency parsing

Features can be defined over

- the next words in the buffer
- the topmost words in the stack
- the partial dependency tree

Training transition-based dependency parsers

- To train a transition-based dependency parser, we need a treebank with dependency trees.
- A collection of freely treebanks for various languages is made available by the Universal Dependencies Project.

  http://universaldependencies.org

Relation extraction

Semantic relations, example

Citing high fuel prices, [ORG United Airlines] said [TIME Friday] it has increased fares by [MONEY $6] per round trip on flights to some cities also served by lower-cost carriers. [ORG American Airlines], a unit of [ORG AMR Corp.], immediately matched the move, spokesman [PER Tim Wagner] said.

is spokesman for

Unified Medical Language

135 entity types, 54 relation types

Injury	**disrupts**	Physiologic Function
Bodily Location	**location-of**	Biologic Function
Anatomical Structure	**part-of**	Organism
Pharmacologic Substance	**causes**	Pathologic Function
Pharmacologic Substance	**treats**	Pathologic Function

Jurafsky and Martin (2017)

Relation extraction using regular expressions

Semantic relations can be extracted using regular expressions.

Example: *\bfödd.*\b

- [PER August Strindberg], född [DATE 22 januari 1849] …

  → \1 was-born-year \2

- [PER August Strindberg], som föddes [DATE 1849], …

  → \1 was-born-year \2

Text patterns for the X is-a Y relation

Pattern	Example
X and other *Y*	… temples, treasuries, **and other** civic buildings.
X or other *Y*	Bruises, wounds, broken bones **or other** injuries …
Y such as *X*	The bow lute, **such as** the Bambara ndang …
Such *Y* as *X*	… **such** authors **as** Herrick, Goldsmith, and Shakespeare.
Y including *X*	… common-law countries, **including** Canada.
Y, especially *X*	European countries, **especially** France and Spain, …

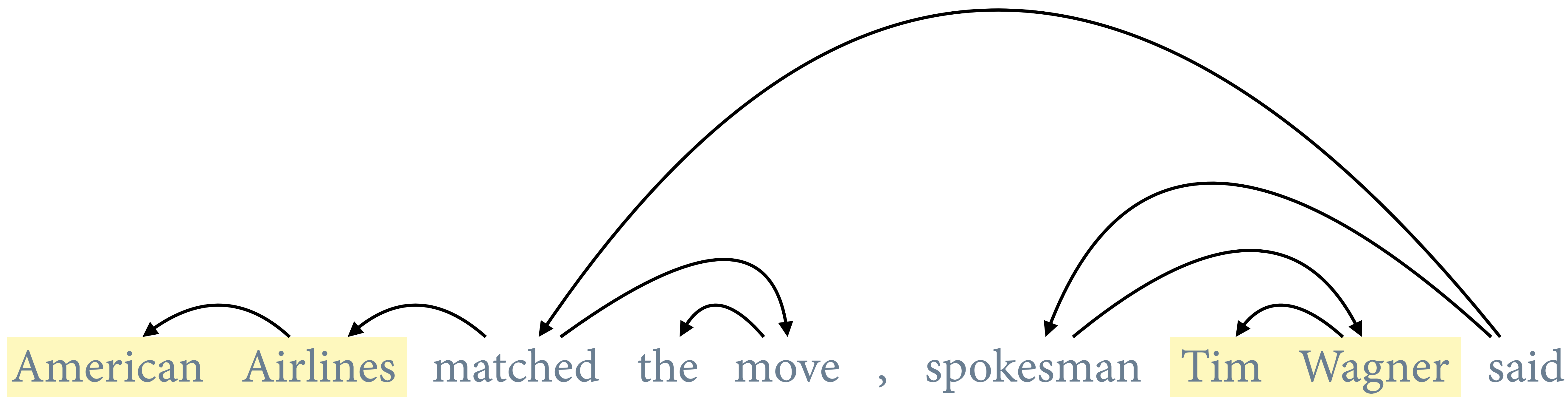
Jurafsky and Martin (2017)

Relation extraction based on dependency trees

- Run the sentence through a named entity recogniser and a dependency parser.
- For each pair of candidate entities, extract the shortest path between the two entities in the tree.
- Feed this path into a neural network and let it predict whether there is a relation between the two entities as well as its type.

  requires recurrent neural networks such as LSTM

Relation extraction based on dependency trees



extracted path between the two entities:
[ORG American Airlines] matched said spokesman [PER Tim Wagner]

Overview of this lecture

- part-of-speech tagging
- named entity recognition
- dependency parsing
- relation extraction