Natural Language Processing

Синтаксический анализ - часть 2

Проект АОТ

Фрагментационный анализ:

- распознавание типа фрагмента;
- применение правил для снятия омонимии;
- применение правил для создания иерархии

Синтаксический анализ: применение правил для составления групп

АОТ: Определение типа фрагмента

Типы фрагментов:

- финитная форма глагола;
- краткое причастие;
- краткое прилагательное;
- предикатив;
- причастие;
- герундий;
- инфинитив;
- вводная фраза;
- null

Пример 1:

`на этот раз она не права`
краткое прилагательное / *null*

Пример 2:

`мои права забрали в милиции`
финитная форма глагола

АОТ: правила для снятия омонимии

1. Если одна из форм является кратким прилагательным или кратким причастием, и нет существительного в Им.п. с тем же самым родом и числом, то этот вариант удаляется: `права он получил только с пятой попытки`;
2. Если есть предикатив без омонимии, то другие варианты омонимичных с предикативом словоформ удаляются: `мыла на кухне она не нашла`

AOT: правила для иерархии

1. **Написанная в спешке** программа выполнила недопустимую операцию. Если причастие стоит перед существительным: ищем существительное с тем же самым падежом, родом, числом. Если существительное найдено, то этот фрагмент включается в следующий и программа выходит из этого правила.
2. Программа, **написанная в спешке,** выполнила недопустимую операцию. Ищем существительное или местоимение с тем же самым родом, числом и падежом. Если слово найдено, то текущий фрагмент включается в следующий и программа выходит из этого правила.

АОТ: Синтаксический анализ

	Пример	Правило	Последовательность	Тип группы

Пример 1	`рубить дрова;`
	`есть кашу;`
	`читать книгу`
		глагол + прямой объект	глагол + существительное (Вин. п.)	глагольная группа

Пример 2	`радостно сообщил;`
	`тихо и смирно ждал;`
	`хорошо знаю`
		наречие + глагол	• наречие + глагол;
			• наречная группа + глагол;
			• наречие + глагольная группа
				глагольная группа

Пример 3	`краше тебя;` `уютнее твоего дома`	сравнительная фраза	две фразы: 1. главное слово - прилагательное в сравнительной форме; 2. главное слово - существительное (Род.п.)	адъективная группа

Link Grammar Parser

https://www.abisource.com/projects/link-grammar/

Этапы анализа:

1. Построение множества всех возможных синтаксических деревьев для предложения;
2. Постобработка синтаксических деревьев предложения.

Link: словари

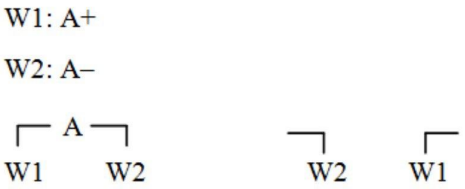
words.n.1	Countable nouns in singular form (`book`)
words.n.2.s	Nouns in plural form ending with “*s*” (`books`)
words.n.2.x	Nouns in plural form not ending with “*s*” (*`man`* - *`men`*)
words.n.3	Uncountable nouns (*`air`*)
words.n.4	Nouns which can be both countable and uncountable (*`tea`*, *`coffee`*)

Link

Connectors: S (subject - predicate), O (object - predicate) etc.

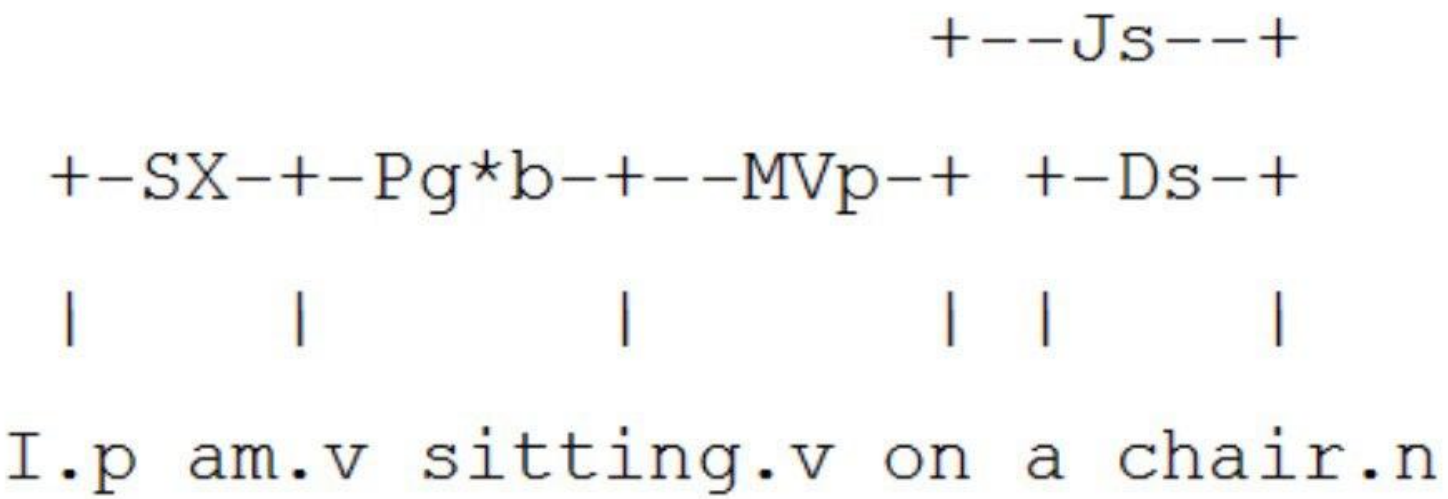
Link directions: “+” - right, “-” - left.

Link = right connector + left connector.



- & - nonsymmetric conjunction. Example: if “*W*: *A*+ & *B*+”, then a word *X*, which has link *A* with word *W*, is located before a word *Y*, which has link *B* with word *W*.
- or - disjunction. If “*W*: *A*+ or *B*-”, then word *W* has right link *A* or left link *B*.
- {} - optionality. If “*W*: *A*+ & {*B*+}”, then after word *W* made right link *A*, it can have or not have link *B*.
- @ - unboundedness

Link



Result of parsing the sentence “*I am sitting on a chair*”

Transition-Based Dependency Parsing

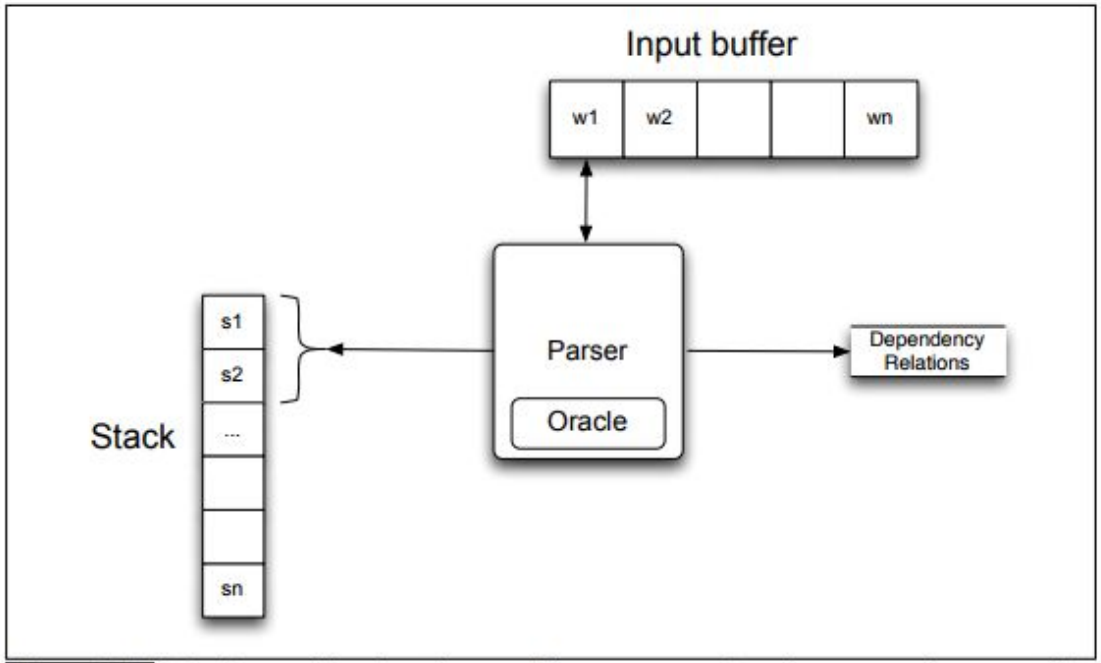


Figure 14.5 Basic transition-based parser. The parser examines the top two elements of the stack and selects an action based on consulting an oracle that examines the current configuration.

Transition-Based Dependency Parsing

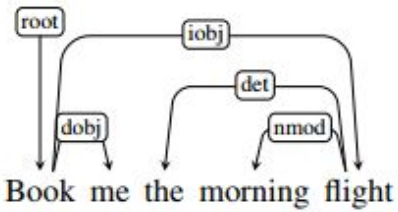
Действия:

- Помечаем текущее слово как главное одного из предыдущих слов;
- Помечаем одно из предыдущих слов как главное текущего слова;
- Ничего не делаем с текущим словом, отправляем его в хранилище для последующей обработки

Операторы перехода:

- **LEFTARC**: Устанавливаем отношение зависимости между первым словом в стэке и следующим за ним словом; второе слово удаляем из стэка;
- **RIGHTARC**: Устанавливаем отношение зависимости между вторым словом и первым словом в стэке; удаляем первое слово в стэке;
- **SHIFT**: Удаляем слово из входного буффера и помещаем его в стэк

Transition-Based Dependency Parsing



Step	Stack	Word List	Action	Relation Added
0	[root]	[book, me, the, morning, flight]	SHIFT	
1	[root, book]	[me, the, morning, flight]	SHIFT	
2	[root, book, me]	[the, morning, flight]	RIGHTARC	(book → me)
3	[root, book]	[the, morning, flight]	SHIFT	
4	[root, book, the]	[morning, flight]	SHIFT	
5	[root, book, the, morning]	[flight]	SHIFT	
6	[root, book, the, morning, flight]	[]	LEFTARC	(morning ← flight)
7	[root, book, the, flight]	[]	LEFTARC	(the ← flight)
8	[root, book, flight]	[]	RIGHTARC	(book → flight)
9	[root, book]	[]	RIGHTARC	(root → book)
10	[root]	[]	Done	

Figure 14.7 Trace of a transition-based parse.

Graph-based Dependency Parsing

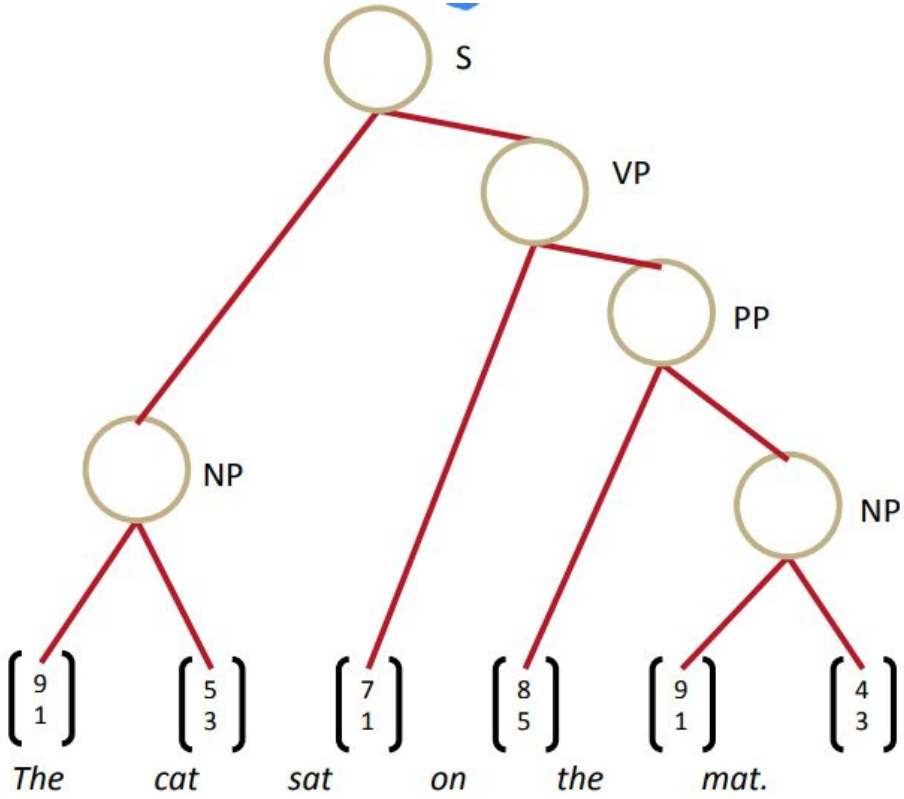
Идея подходов парсинга зависимостей, основанных на графах: в пространстве возможных синтаксических деревьях находим то, которое для входного предложения максимизирует определённое значение.

Формально: для заданного предложения S мы ищем наилучшее дерево зависимостей в G_S - пространстве всех возможных деревьев для этого предложения, которое максимизирует определённое значение.

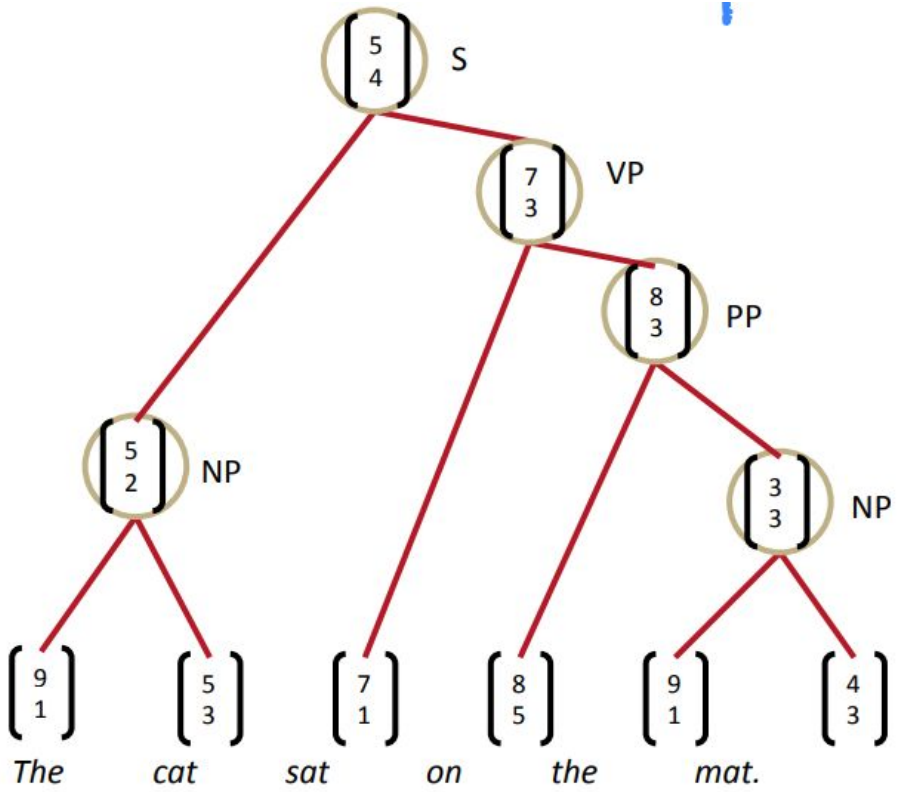
$$\hat{T}(S) = \underset{t \in \mathcal{G}_S}{\operatorname{argmax}}\, score(t, S)$$

$$score(t, S) = \sum_{e \in t} score(e)$$

Recursive Neural Networks

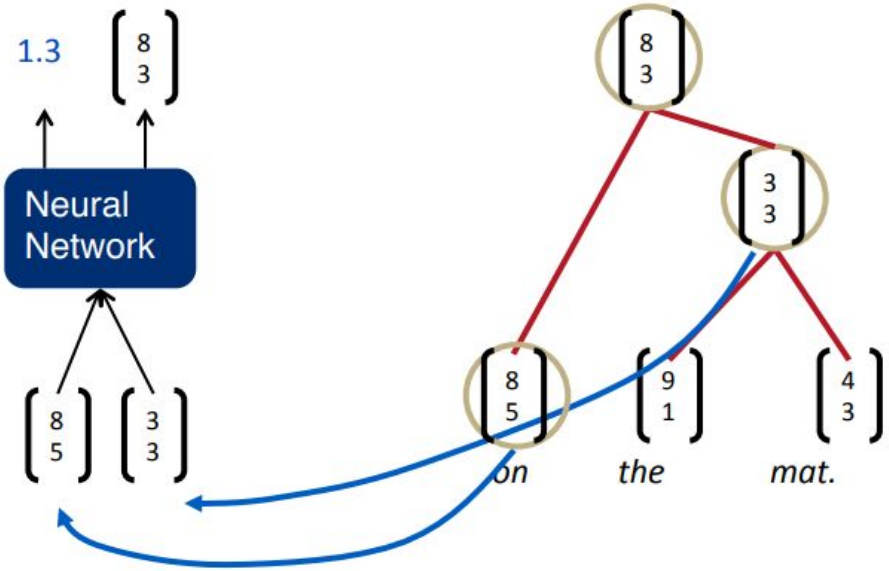


Recursive Neural Networks

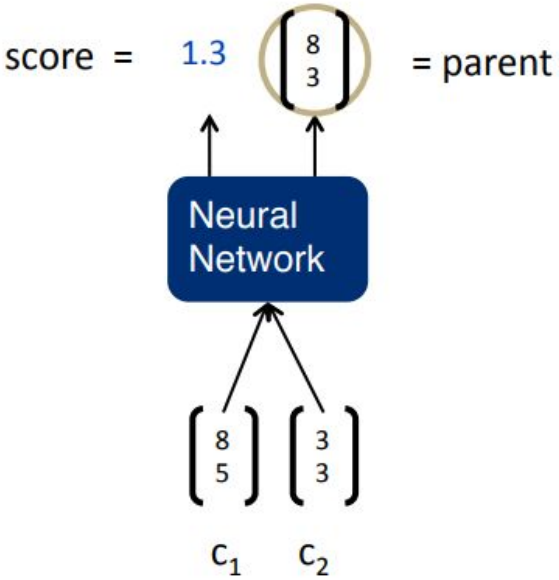


Recursive Neural Networks

- Inputs: two candidate children's representations
- Outputs:
  - 1. The semantic representation if the two nodes are merged.
  - 2. Score of how plausible the new node would be.



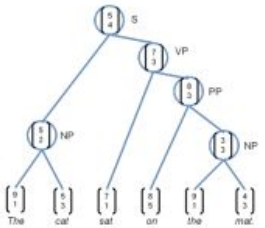
Recursive Neural Networks



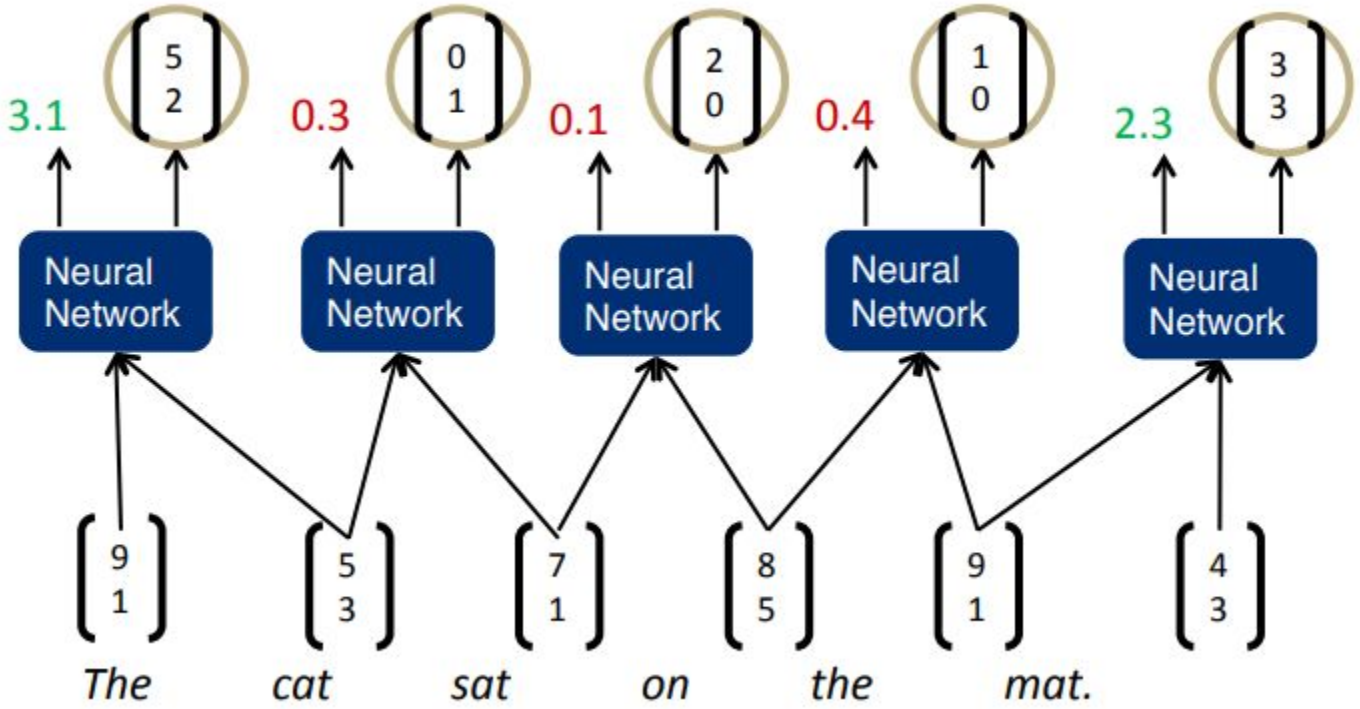
score = $U^T p$

$$p = \tanh\left(W \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} + b\right),$$

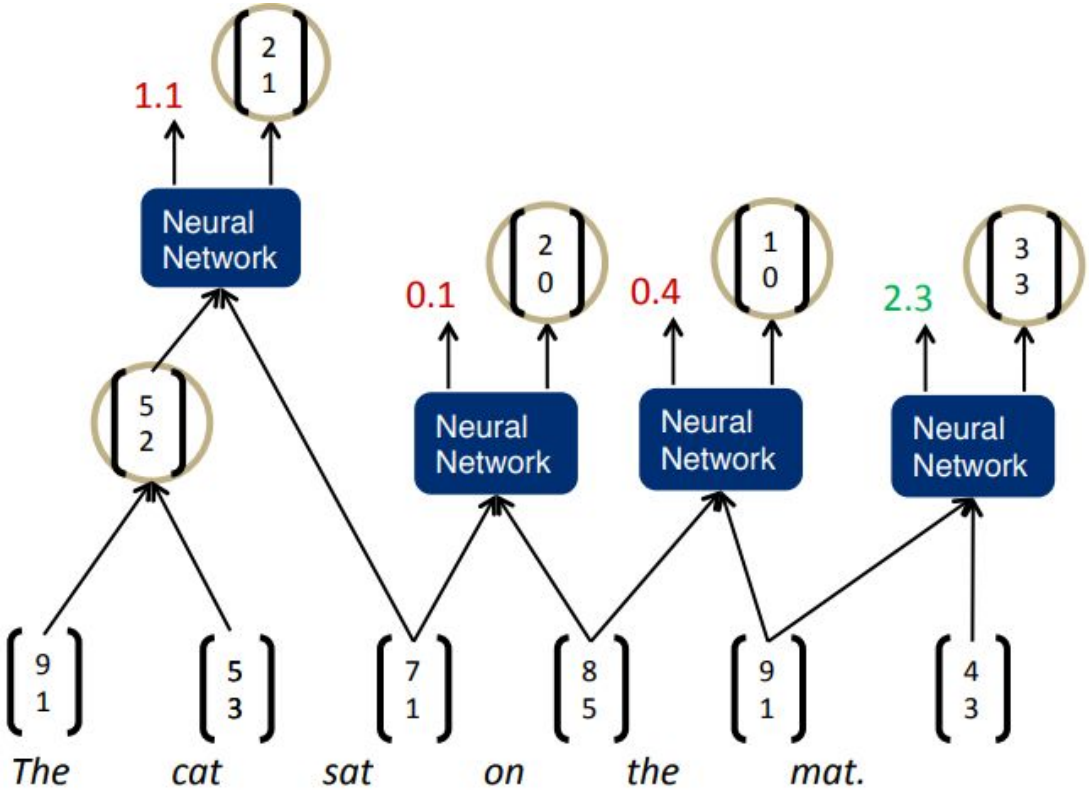
Same W parameters at all nodes of the tree



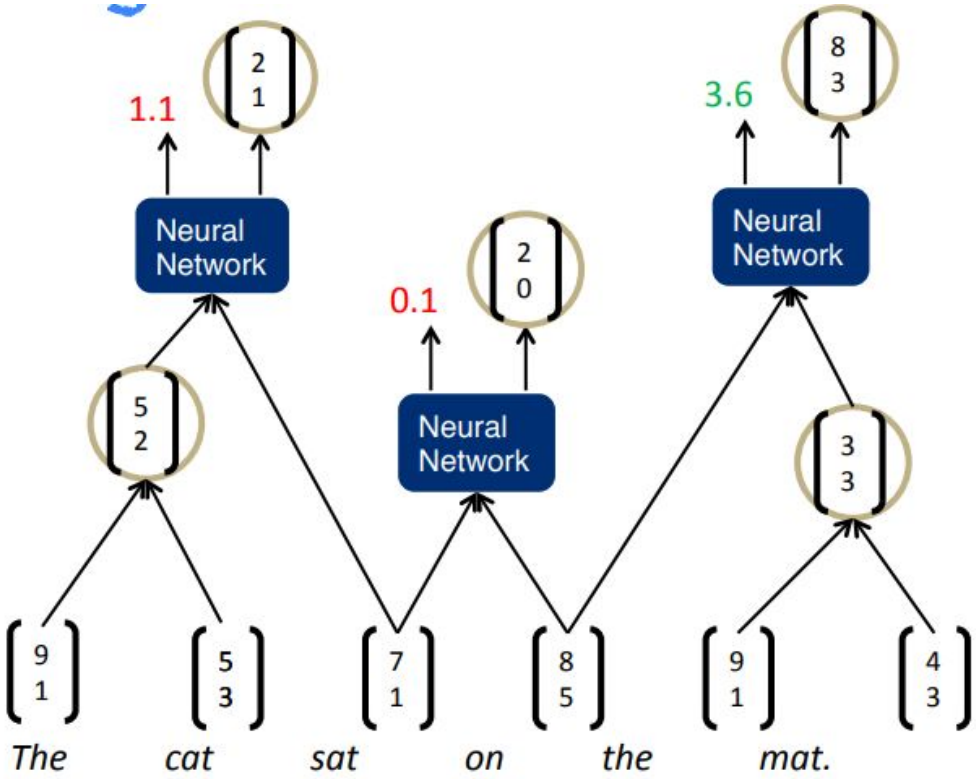
Recursive Neural Networks



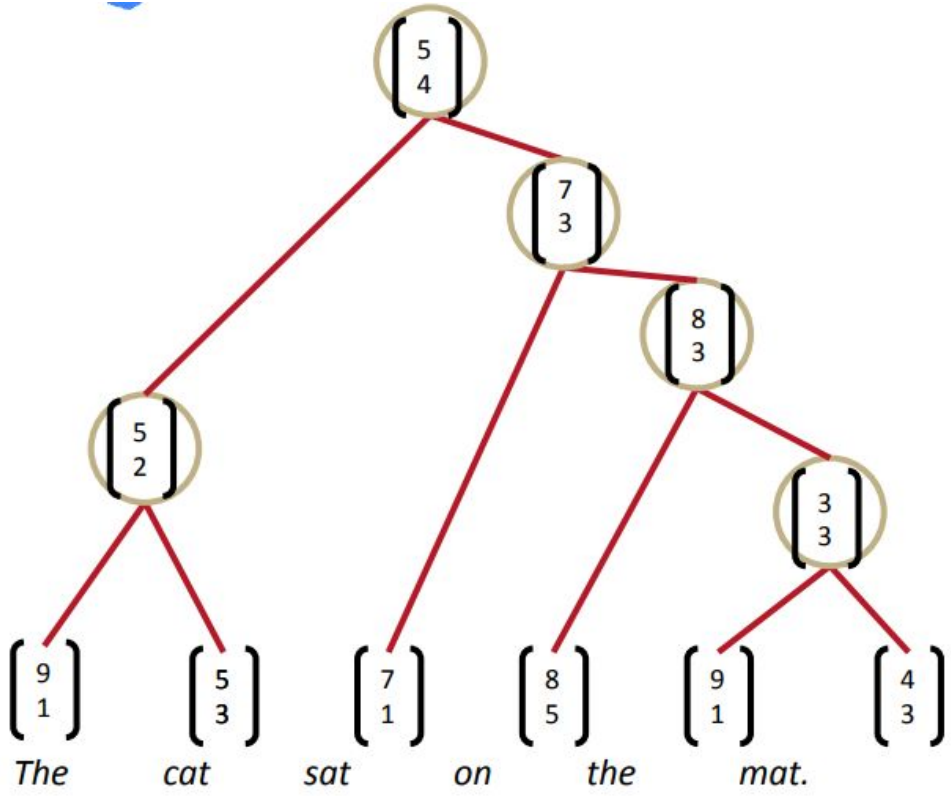
Recursive Neural Networks



Recursive Neural Networks



Recursive Neural Networks



Recursive Neural Networks

We can use it for dependency parsing as well!

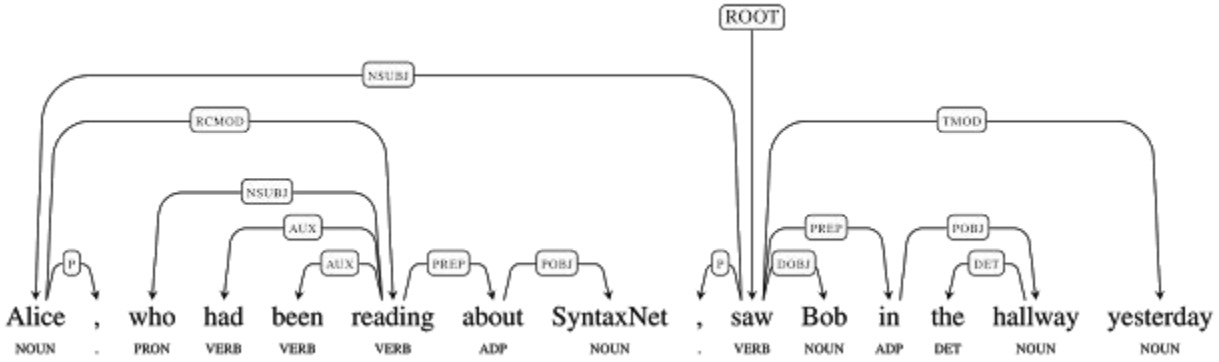
Try out some other features!

More: R.Socher et al. Deep Learning for NLP (without magic): https://nlp.stanford.edu/courses/NAACL2013/NAACL2013-Socher-Manning-DeepLearning.pdf

SyntaxNet

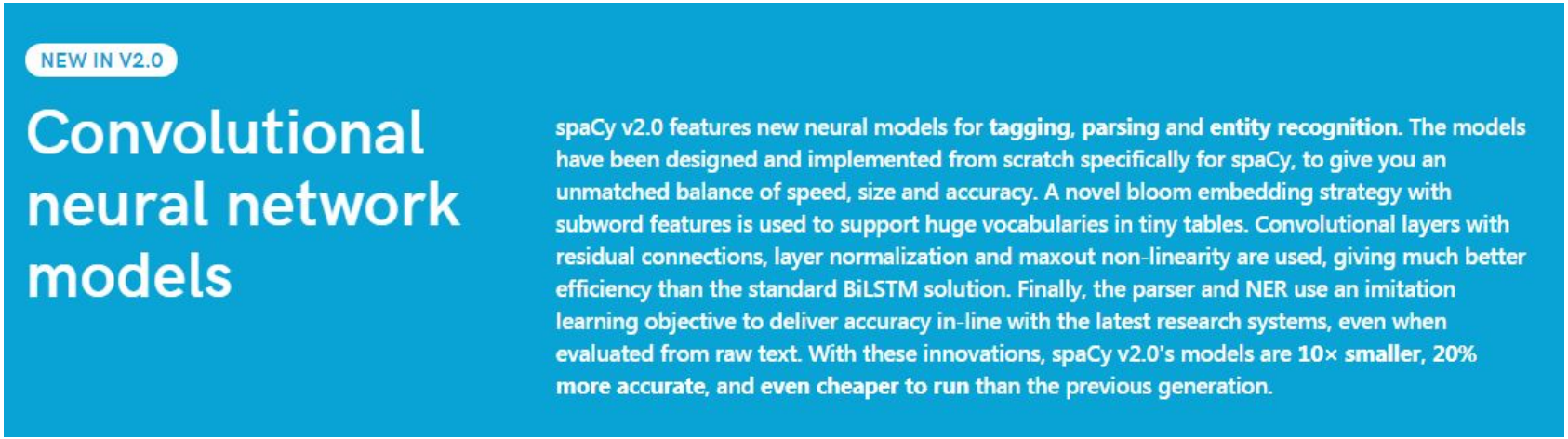
https://research.googleblog.com/2016/05/announcing-syntaxnet-worlds-most.html

Для входного предложения, парсер определяет для каждого слова его часть речи, и определяет синтаксические отношения между словами в предложении, выражая их в дереве зависимостей.



Spacy

https://spacy.io/



Spacy

SYSTEM	YEAR	LANGUAGE	ACCURACY	SPEED (WPS)
spaCy v2.x	2017	Python / Cython	**92.6**	*n/a* ⓘ
spaCy v1.x	2015	Python / Cython	91.8	13,963
ClearNLP	2015	Java	91.7	10,271
CoreNLP	2015	Java	89.6	8,602
MATE	2015	Java	92.5	550
Turbo	2015	C++	92.4	349

Spacy

SYSTEM	YEAR	TYPE	ACCURACY
spaCy v2.0.0	2017	neural	94.48
spaCy v1.1.0	2016	linear	92.80
Dozat and Manning	2017	neural	**95.75**
Andor et al.	2016	neural	94.44
SyntaxNet Parsey McParseface	2016	neural	94.15
Weiss et al.	2015	neural	93.91
Zhang and McDonald	2014	linear	93.32
Martins et al.	2013	linear	93.10

Spacy

	ABSOLUTE (MS PER DOC)			RELATIVE (TO SPACY)		
SYSTEM	TOKENIZE	TAG	PARSE	TOKENIZE	TAG	PARSE
spaCy	**0.2ms**	**1ms**	**19ms**	1x	1x	1x
CoreNLP	2ms	10ms	49ms	10x	10x	2.6x
ZPar	1ms	8ms	850ms	5x	8x	44.7x
NLTK	4ms	443ms	*n/a*	20x	443x	*n/a*

Оценка качества

- **Exact match** (EM) — количество предложений, которые были разобраны корректно;
- **Labeled attachment score** (LAS) - количество слов, для которых было правильно определено главное слово и синтаксическое отношение;
- **Unlabeled attachment score** (UAS) - количество слов, для которых было правильно определено главное слово, синтаксическое отношение игнорируется;
- **Label accuracy score** (LS) - процент токенов, для которых правильно определён тип синтаксического отношения, игнорируя сами отношения;
- Насколько система хорошо определяет конкретные типы отношений, например, *NSUBJ*, во всём корпусе - использовать precision и recall !

Evaluation

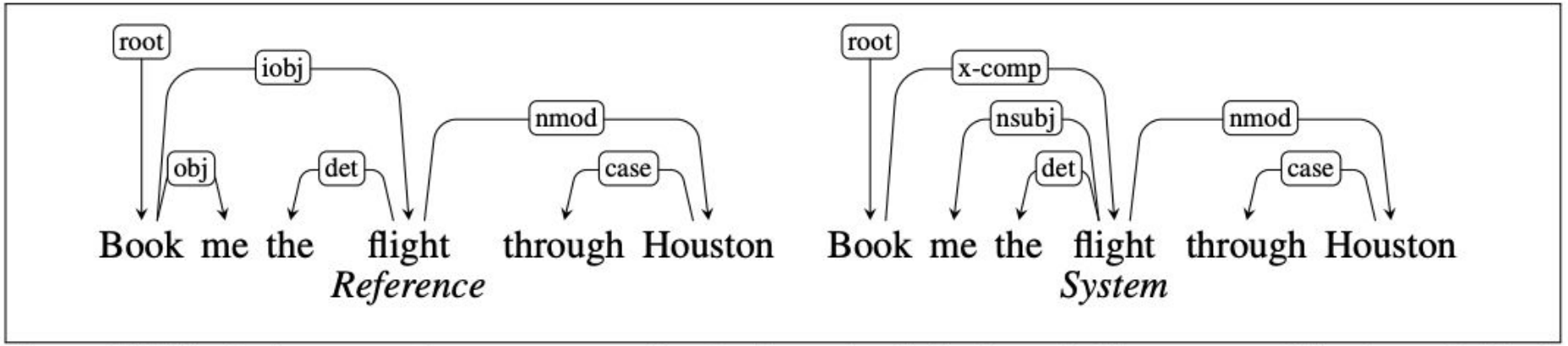
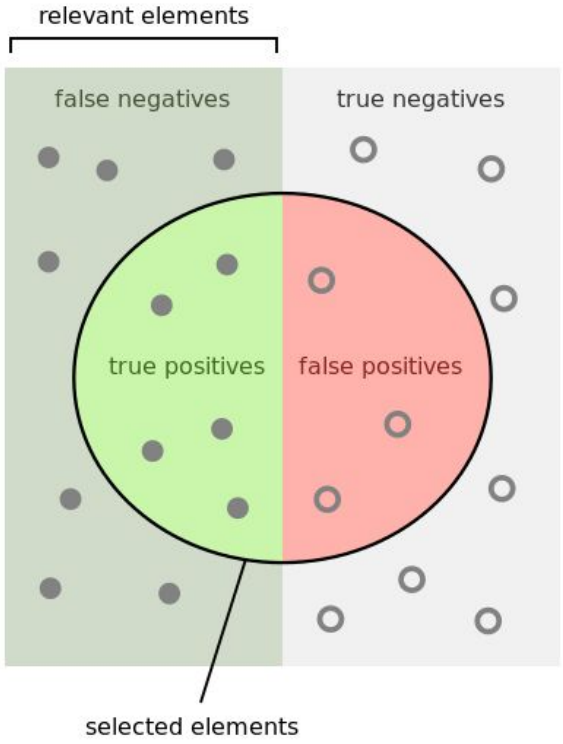


Figure 15.15 Reference and system parses for *Book me the flight through Houston*, resulting in an LAS of 2/3 and an UAS of 5/6.

Precision, Recall, F-measure



How many selected items are relevant?

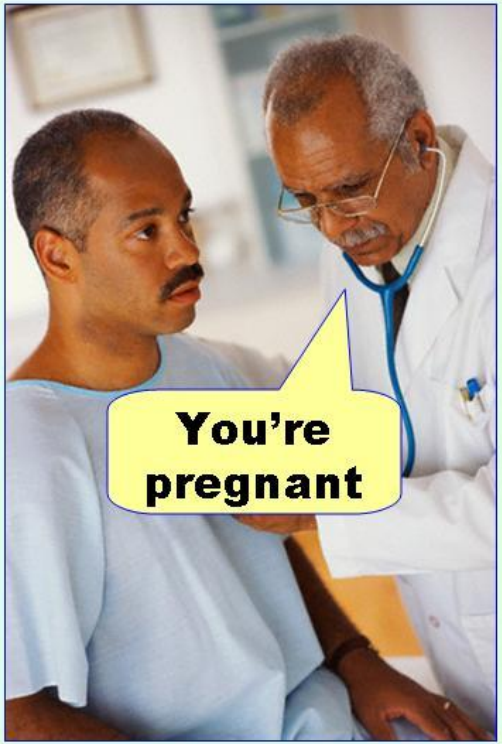
$$\text{Precision} = \frac{\text{true positives}}{\text{true positives} + \text{false positives}}$$

How many relevant items are selected?

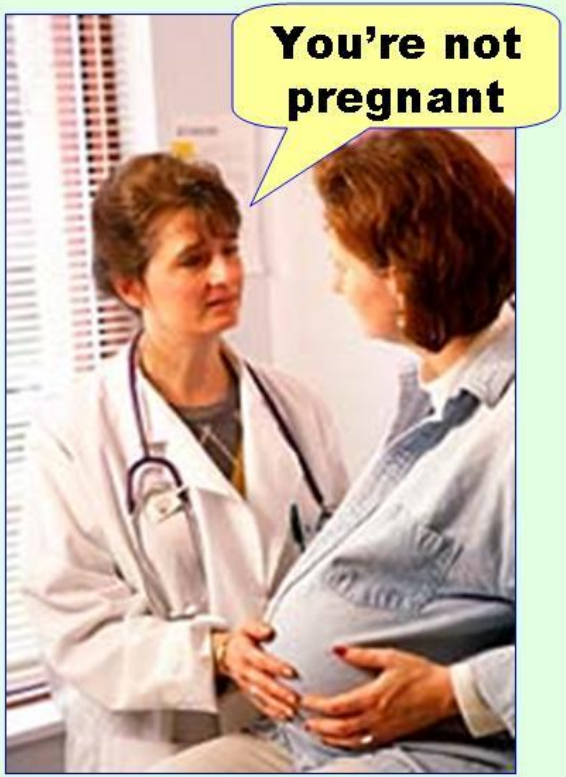
$$\text{Recall} = \frac{\text{true positives}}{\text{true positives} + \text{false negatives}}$$

$$F1 = 2 * \frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}}$$

Type I error
(false positive)



Type II error
(false negative)



Сравнение синтаксических анализаторов для русского языка (2012)

Parser	P	R	F1
Compreno	0.952	0.983	0.967
ETAP-3	0.933	0.981	0.956
SyntAutom	0.895	0.980	0.935
SemSyn	0.889	0.947	0.917
Dictum	0.863	0.980	0.917
Semantic Analyzer Group	0.856	0.860	0.858
AotSoft	0.789	0.975	0.872

DeepPavlov.ai

Dataset	Model	UAS	LAS
UD2.3 (Russian)	UD Pipe 2.3 (Straka et al., 2017)	90.3	89.0
	UD Pipe Future (Straka, 2018)	93.0	91.5
	UDify (multilingual BERT) (Kondratyuk, 2018)	94.8	93.1
	our BERT model	95.2	93.7

So our model is the state-of-the-art system for Russian syntactic parsing by a valuable margin.

State-of-the-art syntax parsers

http://nlpprogress.com/english/dependency_parsing.html

Model	POS	UAS	LAS	Paper / Source	Code
Label Attention Layer + HPSG + XLNet (Mrini et al., 2019)	97.3	97.42	96.26	Rethinking Self-Attention: Towards Interpretability for Neural Parsing	Official
HPSG Parser (Joint) + XLNet (Zhou and Zhao, 2019)	97.3	97.20	95.72	Head-Driven Phrase Structure Grammar Parsing on Penn Treebank	Official
HPSG Parser (Joint) + BERT (Zhou and Zhao, 2019)	97.3	97.00	95.43	Head-Driven Phrase Structure Grammar Parsing on Penn Treebank	Official
CVT + Multi-Task (Clark et al., 2018)	97.74	96.61	95.02	Semi-Supervised Sequence Modeling with Cross-View Training	Official

Thank you for your attention!