(Probably) Everything You Need to Know About Dependency Parsing

What is parsing? Why do we need it?

- We need to put structures over sentences to understand/make computers understand how the words in the sentence relate to one another.

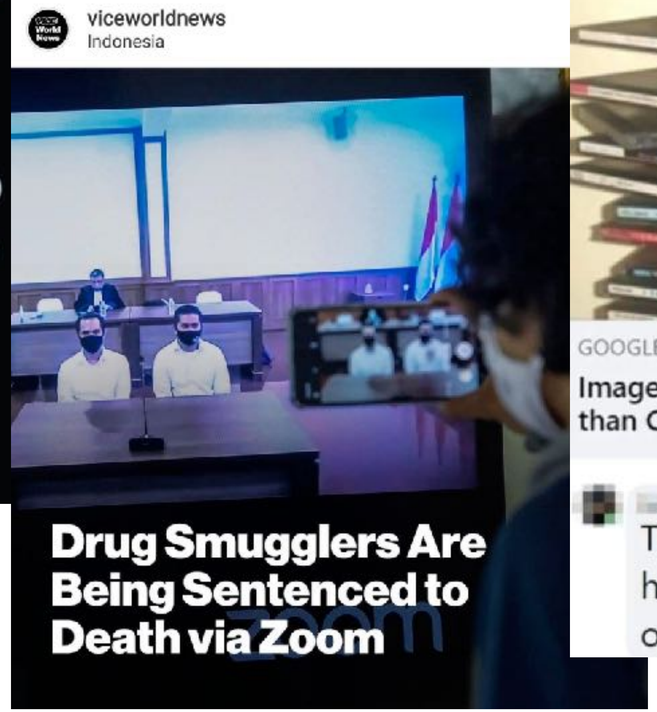
  The girl killed the bear. -vs- The bear killed the girl.

- So, parsing is practically determining the grammar structure of an input/ sentence.

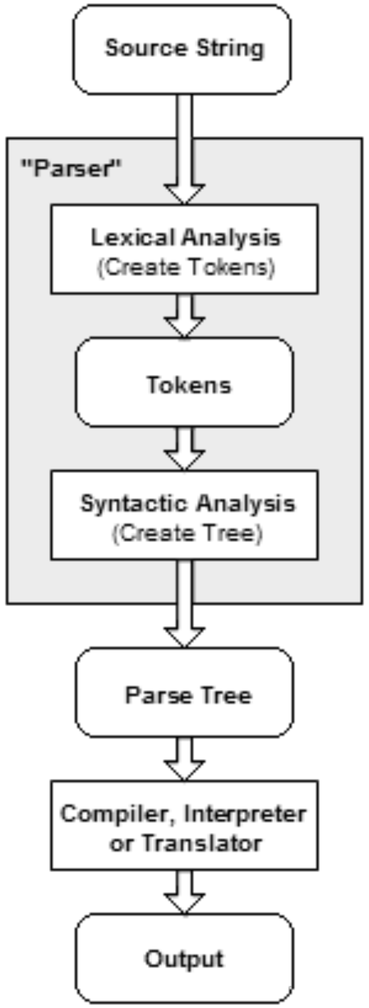
- A fancier definition: "*Analysing a text, made of a sequence of tokens (for example, words), to determine its grammatical structure within the framework of a (formal) grammar.*"

What is parsing? Why do we need it?

- Parsing is not a very easy task for computers since human languages can be VERY ambiguous. So, cmpe people, nlp people and linguists came up with different strategies for parsing.

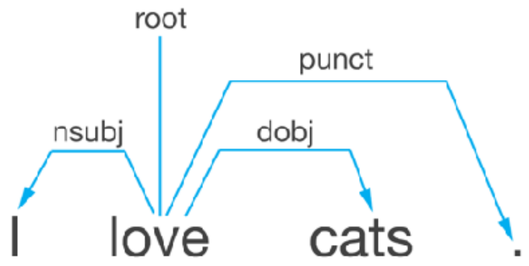


What is parsing? Why do we need it?



“I love cats.”

I, love, cats, .



Tiny bit of theoretical background on dependency grammars

Phrase structure grammar

- Chomsky.
- Based on the notion of **constituency relations.**
    - Dates back to Aristotle & term logic.
    - Subject - predicate division.
- Binary branching and binary division. (remember X' theory)
- Almost everything we learned in theory so far belongs to this category. (Govt & binding, minimalist program, nanosyntax, phrase structure grammar…)

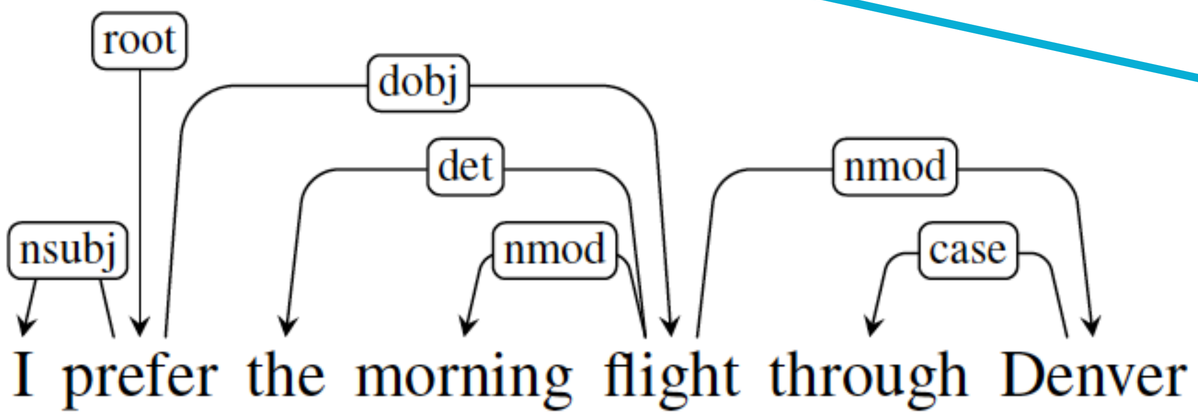
Tiny bit of theoretical background on dependency grammars

Dependency grammar

- Tesnière.
- Based on the notion of **dependency relations.**
    - Defined using notions of head & dependent.
    - Linguistic units are connected to one another w/ links.
- Verb (predicate) is the king. Everything is connected to it directly or indirectly.
- Flatter. (no bar levels, no phrase levels etc.)

What is dependency parsing?

- Dependency parsing is based on dependency grammar.
- It illustrates relation between heads and their dependents using a set of **predefined tags.**



Mainly based on
POS tags

- Arrow implies head & its dependent, tag shows the relation.

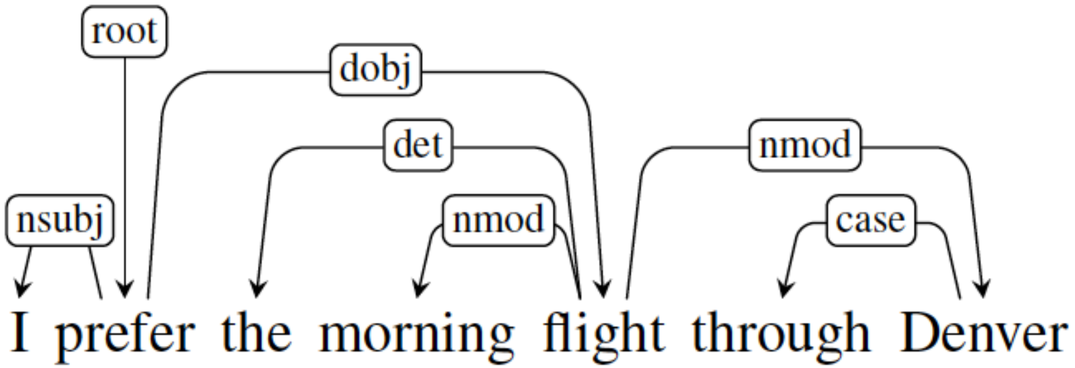
POS Tags & Dependency Tags

Open class words	Closed class words	Other
ADJ	ADP	PUNCT
ADV	AUX	SYM
INTJ	CONJ	X
NOUN	DET	
PROPN	NUM	
VERB	PART	
	PRON	
	SCONJ	

	Nominals	Clauses	Modifier words	Function Words
Core arguments	nsubj obj iobj	csubj ccomp xcomp		
Non-core dependents	obl vocative expl dislocated	advcl	advmod* discourse	aux cop mark
Nominal dependents	nmod appos nummod	acl	amod	det clf case
Coordination	**MWE**	**Loose**	**Special**	**Other**
conj cc	fixed flat compound	list parataxis	orphan goeswith reparandum	punct root dep

Basic features of a dependency tree

- Each lemma has only one incoming arrow. Not zero, not two.
- Lemmas can have zero or multiple outgoing arrows.
- The predicate is the root. (“Lexical verb”)
- Function words cannot be heads. (Prepositions, articles, auxiliaries…)
- There must be a unique path between the root and each lemma.

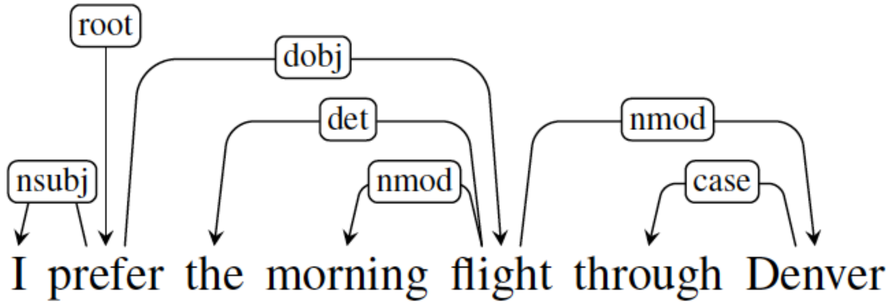


Some terminology

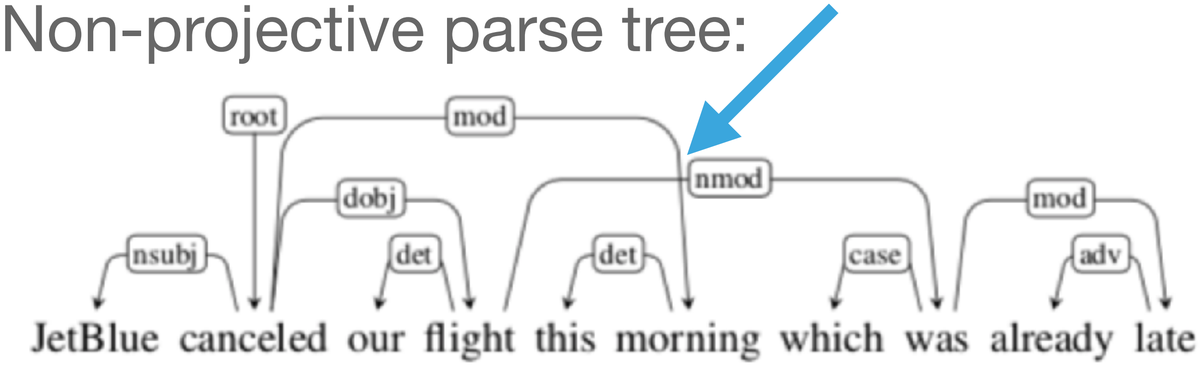
- ROOT: The root of the dependency tree. (like S)
- LEMMA: Lemma or stem of word form.
- UPOSTAG: Universal part-of-speech tag.
- XPOSTAG: Language-specific part-of-speech tag.

Some terminology

- Projective parse tree:

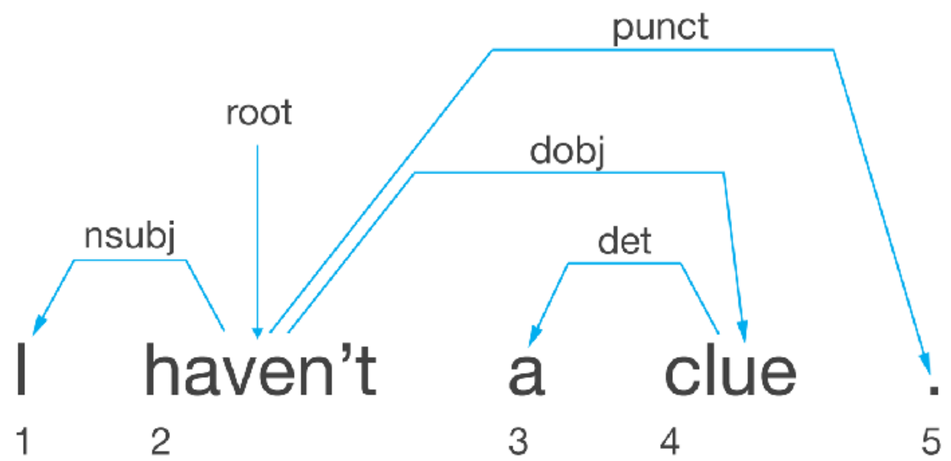


- Non-projective parse tree:

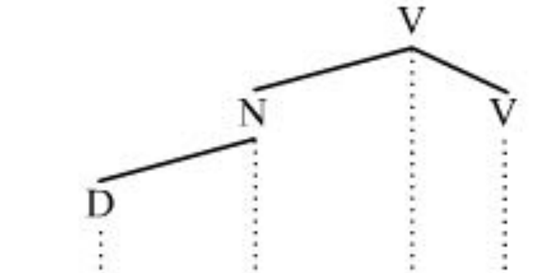


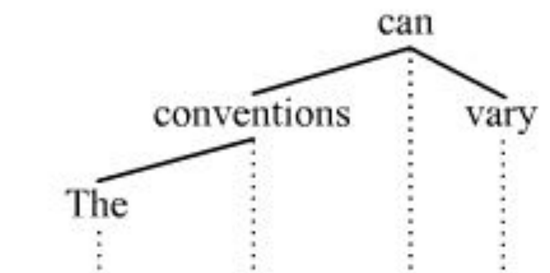
An actual dependency annotation

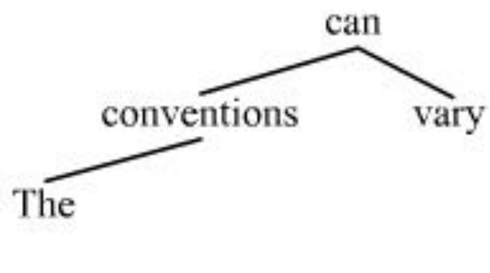
```
1     I          I       PRON     PRP    Case=Nom|Number=Sing|Person=1                  2    nsubj
2     haven't    _       VERB     _      Negative=Neg|Number=Sing|Person=1|Tense=Pres    0    root
3     a          a       DET      DT     Definite=Ind|PronType=Art                      4    det
4     clue       clue    NOUN     NN     Number=Sing                                    2    dobj
5     .          .       PUNCT    .      _                                              2    punct
```

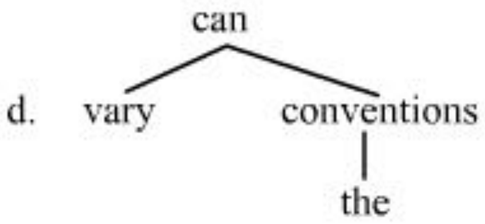


Different visual representations of dependency trees

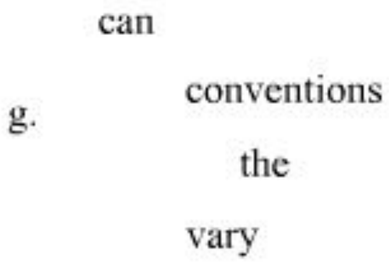








f. [[The] conventions] can [vary].



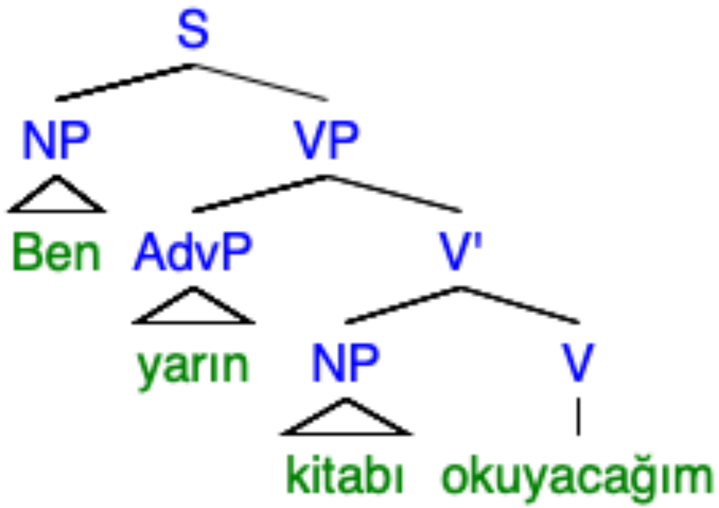
Dependency vs. Constituency trees

- Constituency trees were very popular in the past.
- Challenges for constituency trees: Projectivity, morphologically rich languages, free word order languages etc.
- Dependency trees are able to handle such challenges better. Famously, dependency models fit free word order languages much better.
- For a thorough discussion of dependency & constituency parsing in terms of parser performance: https://www.aclweb.org/anthology/P04-1061.pdf

Dependency vs. Constituency trees

“Kitabı yarın ben okuyacağım.”

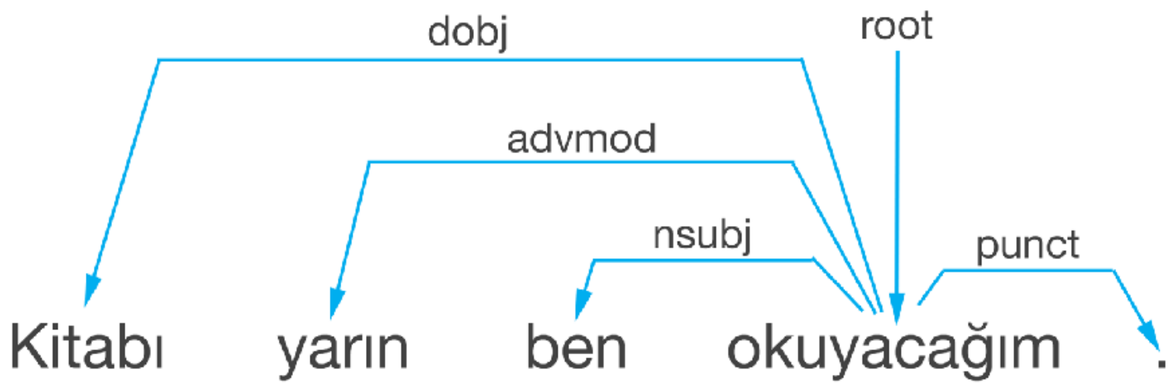
To derive this, we need tons of movement & operations. The end result will be extremely complex even if we stick to the very reduced & simple trees from Ling101.



Dependency vs. Constituency trees

“Kitabı yarın ben okuyacağım.”

Yet this is not a challenge for dependency trees:



Towards dependency trees: Route 1

Annotations

- Interfaces:
    - UD Annotatrix
      https://github.com/jonorthwash/ud-annotatrix
    - Arborator
      https://arborator.ilpga.fr/q.cgi
    - WebAnno
      https://webanno.github.io/webanno/
    - Conllu Editor
      https://github.com/Orange-OpenSource/conllueditor

Towards dependency trees: Route 2

Parsers

- Tools:
    - Stanza
      https://github.com/stanfordnlp/stanza
    - NLTK
      https://www.nltk.org
    - SpaCy
      https://github.com/explosion/spaCy

Towards dependency trees: Route 3

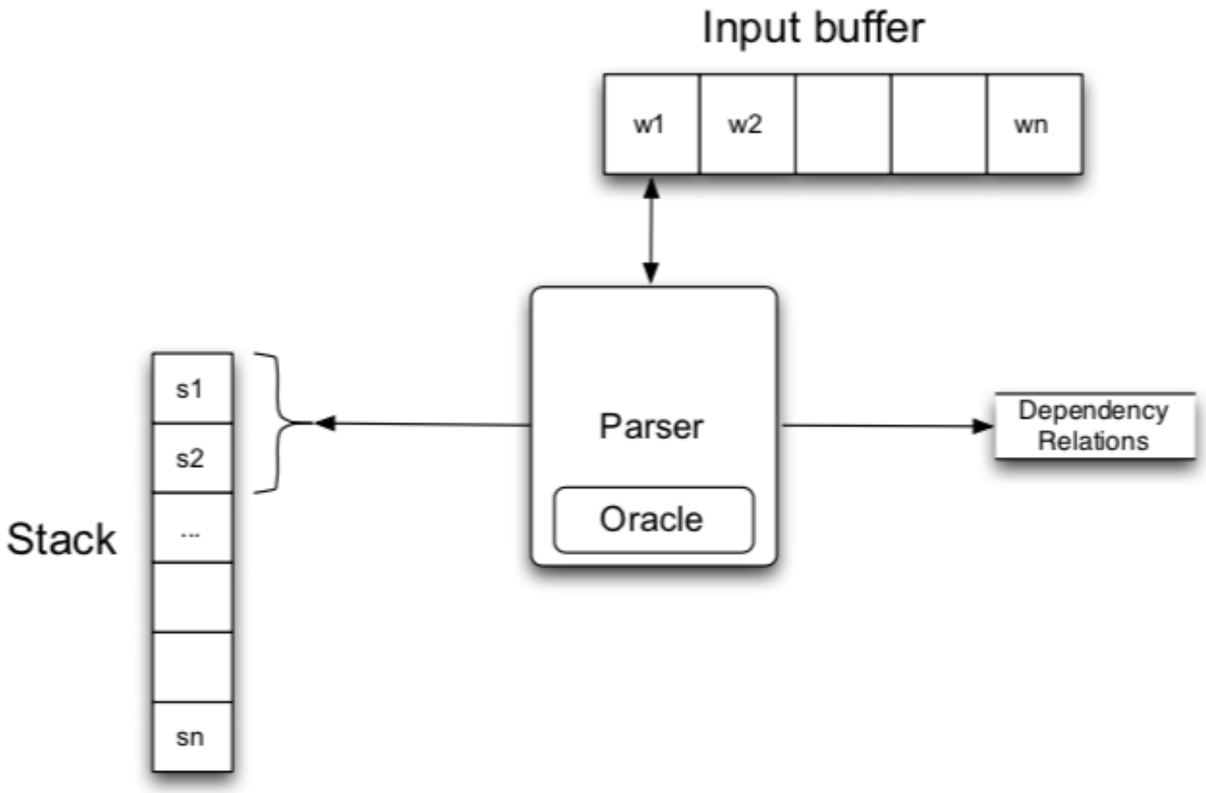
Making Our Own Parser

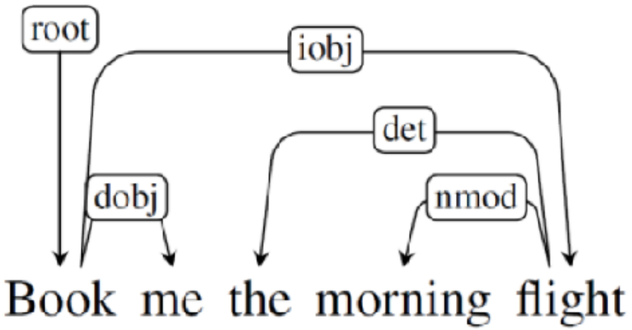
- Shift reduce parsing
- Graph-based parsing
- Maximum spanning tree
- ....

Towards dependency trees: Route 3

Shift reduce parsing

- The most straightforward one. Uses 3 transition operations.

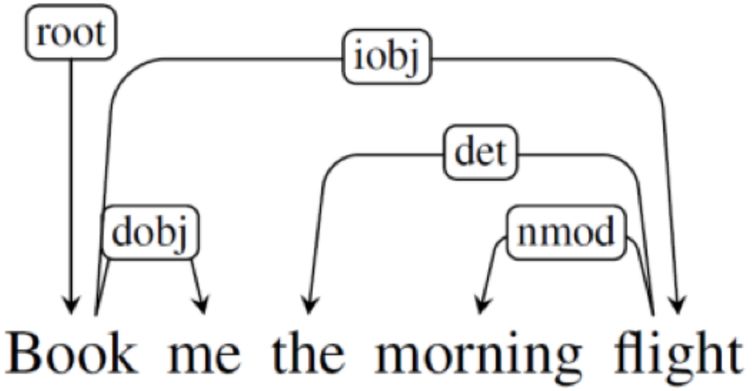




Towards dependency trees: Route 3

Shift reduce parsing

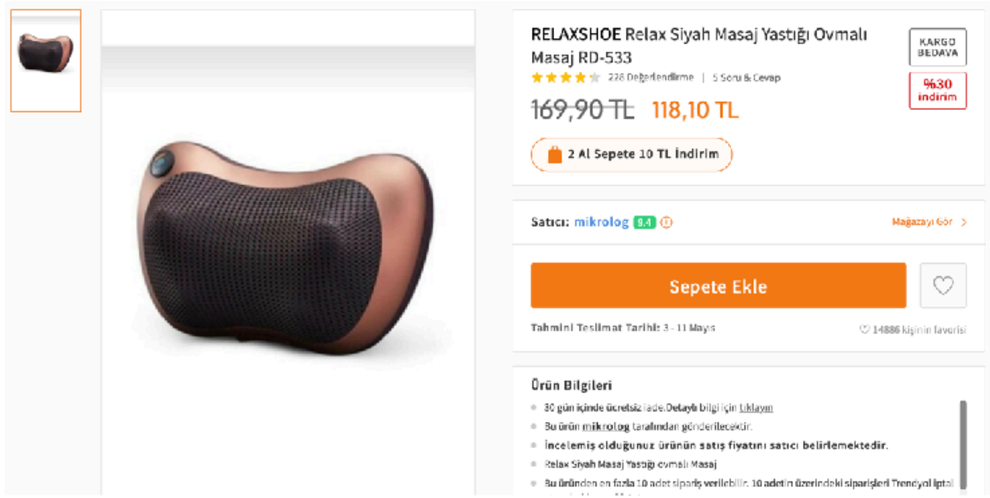
Step	Stack	Word List	Action	Relation Added
0	[root]	[book, me, the, morning, flight]	SHIFT	
1	[root, book]	[me, the, morning, flight]	SHIFT	
2	[root, book, me]	[the, morning, flight]	RIGHTARC	(book → me)
3	[root, book]	[the, morning, flight]	SHIFT	
4	[root, book, the]	[morning, flight]	SHIFT	
5	[root, 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	



Use cases of dependency parsing, #1

- Many industries heavily rely on customer feedback.
- Hiring people to read all customer comments/mails, note frequent and important issues etc. is often just wasteful. That is why automating such processes is very efficient and becoming popular.

Use cases of dependency parsing, #1



RELAXSHOE Relax Siyah Masaj Yastığı Ovmalı Masaj RD-533

KARGO BEDAVA

238 Değerlendirme | 5 Soru & Cevap

%30 indirim

169,90 TL 118,10 TL

2 Al Sepete 10 TL İndirim

Satıcı: mikrolog 9.4 Mağazayı Gör

Sepete Ekle

Tahmini Teslimat Tarihi: 3 - 11 Mayıs

14886 kişinin favorisi

Ürün Bilgileri

- 30 gün içinde ücretsiz iade. Detaylı bilgi için Tıklayın
- Bu ürün mikrolog tarafından gönderilecektir.
- İncelemiş olduğunuz ürünün satış fiyatını satıcı belirlemektedir.
- Relax Siyah Masaj Yastığı ovmalı Masaj
- Bu üründen en fazla 10 adet sipariş verilebilir. 10 adetin üzerindeki siparişleri Trendyol iptal

★★★★★ gerçekten harika bir ürün kesinlikle almalısınız

sibel sever - Elite Üye | 8 Nisan 2021 | mikrolog satıcısından alındı. Ürünü satın aldı

Yorumu Beğen (0)

★★★☆☆ ürün beklenildigi kadar güçlü değil ayağımın ağırlığına bile kitleniyor 15 dkdan fazla kullanamiyorsun ısınma yapıyor

**** **** | 4 Nisan 2021 | mikrolog satıcısından alındı. Ürünü satın aldı

Yorumu Beğen (0)

★★★☆☆ daha önce çevremede önererek 10 adet aldım. kendimde olanı aileme verip yeni sipariş ettim fakat son aldığım ürün güçsüz ve sıkıntılı çıktı ve trendyol hızlı iade ile iade ettim , eğer sağlam ürün gelirse kesinlikle öneririm

S** U** | 6 Nisan 2021 | mikrolog satıcısından alındı. Ürünü satın aldı

Yorumu Beğen (0)

★★★★☆ fiyatına göre iyi bir ürün biz beğendik gerçekten özellikle sırt kısmını rahatlatıcı bir etkisi var tek eksi yani kablosu kısa üçlü kullanmak zorundasiniz

Esra Ilmez | 1 Mayıs 2021 | Chermik satıcısından alındı. Ürünü satın aldı

Yorumu Beğen (0)

★★★★☆ Biraz daha güçlü motor olsa 10 numara olurmuş aslında

M** B** | 24 Nisan 2021 | mikrolog satıcısından alındı. Ürünü satın aldı

Yorumu Beğen (2)

★★★★★ Fiyatına göre performansı gayet iyi. Kulunçları hemen buluyor çok başarılı. Tabi bu fiyata aşırı kaliteli bir ürün beklemeyin fiyat performans ürünü memnunuz ailece.

C** C** | 29 Nisan 2021 | mikrolog satıcısından alındı. Ürünü satın aldı

Yorumu Beğen (0)

Use cases of dependency parsing, #1

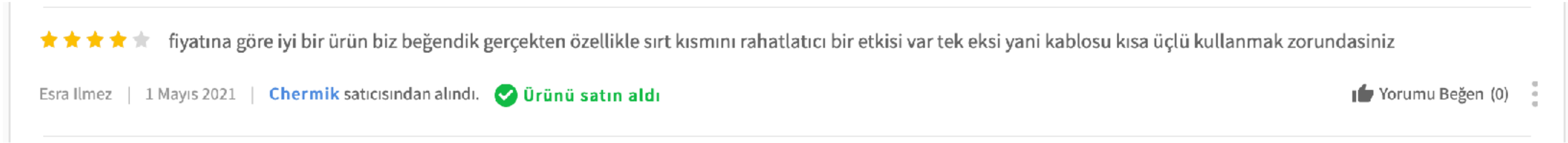
- This product has 150+ comments. Imagine having 10 products like this & trying to keep up with customer feedback swarming in from multiple sources: Hepsiburada, Trendyol, Amazon, N11, your own website…

- A way to process all comments: Search for some keywords & count them. “harika,” “güçlü,” “sağlam ürün”

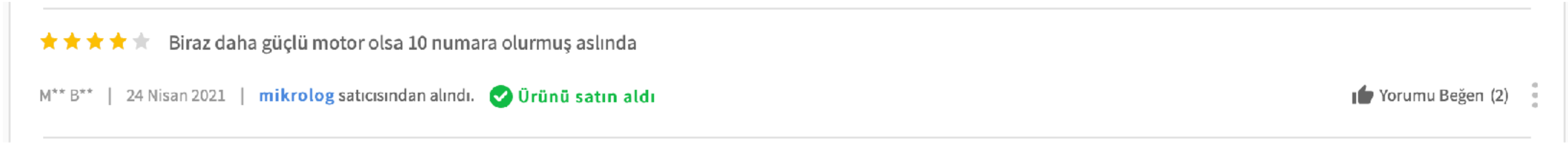
  Can be misleading!

Use cases of dependency parsing, #1

- Are “fiyatına göre iyi” and “iyi” the same?



- Or “güçlü motor” and “güçlü motor olsa” ?



Use cases of dependency parsing, #1

- Not really. We can distinguish such comments from one another by parsing these sentences.

- We can also distinguish:
  “Kötü bir ürün değil.” - “Kötü bir ürün.”
  “X değil Y olsa muhteşem.” - “Muhteşem.”

Use cases of dependency parsing, #2

- Do you use a voice assistant? I do!

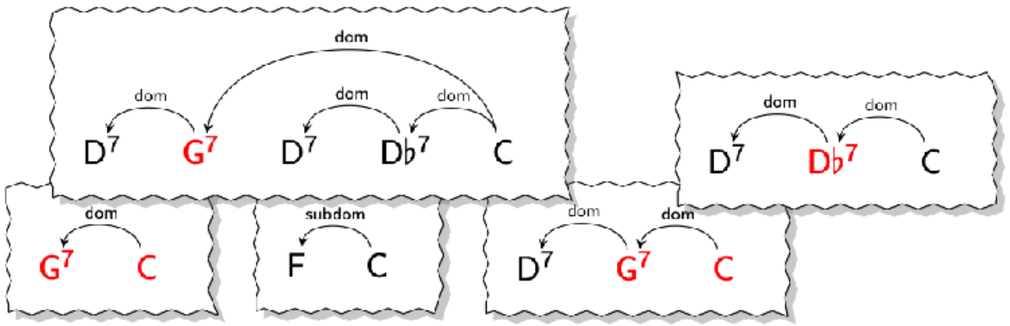
- How do they distinguish commands like
  “Siri, anneme gidicem yaz.” and “Siri, anneme gidicem.”

- Yes, again, parsing! Some voice assistants use dependency parsing, some use shallow parsing.

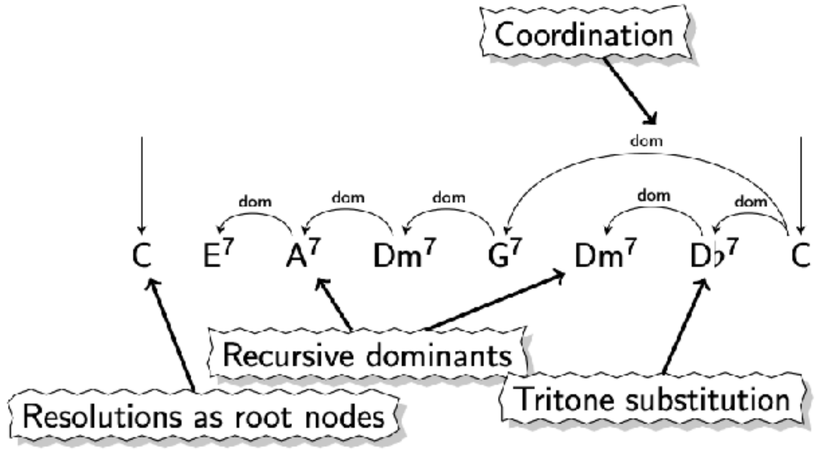
Trivia: Dependency grammar can also be used in music theory!

Harmonic Analysis

- Chords classified as functioning as: dominant, subdominant or tonic
- Dominant-tonic resolution
- Subdominant-tonic resolution
- Recursion
- Substitution
- Delayed resolution: coordination



Harmonic Analysis Dependency Graph



http://mark.granroth-wilding.co.uk/files/mml2012_article.pdf

Dependency Treebanks

- UD Treebank
  https://github.com/UniversalDependencies
- UD Turkish BOUN
  https://github.com/boun-tabi/UD_Turkish-BOUN
- UD Penn Turkish
  https://github.com/UniversalDependencies/UD_Turkish-Penn

Resources

- More on DP and conversions b/w constituency & dependency: https://web.stanford.edu/~jurafsky/slp3/14.pdf
- Detailed explanations of dependency tags: https://universaldependencies.org/u/dep/
- Stanford Dependencies: https://nlp.stanford.edu/software/stanford-dependencies.shtml
- Universal Dependencies: https://universaldependencies.org
- CoNNL format guide: https://universaldependencies.org/docs/format.html
- UD Tools https://universaldependencies.org/tools.html#arborator