

x

-

(<http://play.google.com/store/apps/details?id=com.analyticsvidhya.android>)



LOGIN / REGISTER ([HTTPS://ID.ANALYTICSVIDHYA.COM/ACCOUNTS/LOGIN/](https://id.analyticsvidhya.com/accounts/login/))

NEXT=[HTTPS://WWW.ANALYTICSVIDHYA.COM/BLOG/2017/12/INTRODUCTION-COMPUTATIONAL-LINGUISTICS-DEPENDENCY-TREES/](https://www.analyticsvidhya.com/blog/2017/12/introduction-computational-linguistics-dependency-trees/))



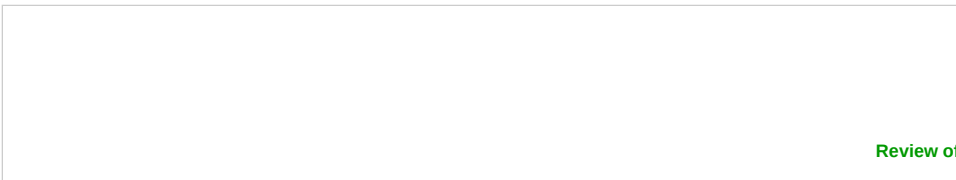
(https://datahack.analyticsvidhya.com/contest/capillary-machine-learning-hackathon/?utm_source=bannerbelowAVblogtitle)

[NLP \(HTTPS://WWW.ANALYTICSVIDHYA.COM/BLOG/CATEGORY/NLP/\)](https://www.analyticsvidhya.com/blog/category/nlp/)

[PYTHON \(HTTPS://WWW.ANALYTICSVIDHYA.COM/BLOG/CATEGORY/PYTHON-2/\)](https://www.analyticsvidhya.com/blog/category/python-2/)

Introduction to Computational Linguistics and Dependency Trees in data science

SHIVAM BANSAL ([HTTPS://WWW.ANALYTICSVIDHYA.COM/BLOG/AUTHOR/SHIVAM5992/](https://www.analyticsvidhya.com/blog/author/shivam5992/)), DECEMBER 14, 2017



Introduction

In recent years, the amalgam of deep learning fundamentals with Natural Language Processing techniques has shown a great improvement in the information mining tasks on unstructured text data.

Subscribe!

Download Brochure

The models are now able to recognize natural language and speech comparable to human levels. Despite such improvements, discrepancies in the results still exist as sometimes the information is coded very deep in the syntaxes and syntactic structures of the corpus.



Example – Problem with Neural Networks

For example, a conversation system which is trained using recurrent neural network produces the following results in two scenarios:

User: Hi, I took a horrible picture in a museum, can you tell where is it located?

Bot Reply 1: The museum is located at a horrible place

User: Hi, I took a horrible picture in a museum, can you tell where is it located?

Bot Reply 2: The horrible museum is located at this place

The two responses have virtually similar tokens but different structures which completely changes the context. In this article, I will discuss the interdisciplinary field of Computational Linguistics which deals with the structural aspects of the text that are used to solve common text related problems. Some examples are Named entity extraction, coreference resolution, and machine translation.

Computational linguistics

Computational linguistics often overlaps with the field of Natural Language Processing as most of the tasks are common to both the fields. While Natural Language Processing focuses on the tokens/tags and uses them as predictors in machine learning models, Computational Linguistics digs further deeper into the relationships and links among them.

Subscribe!

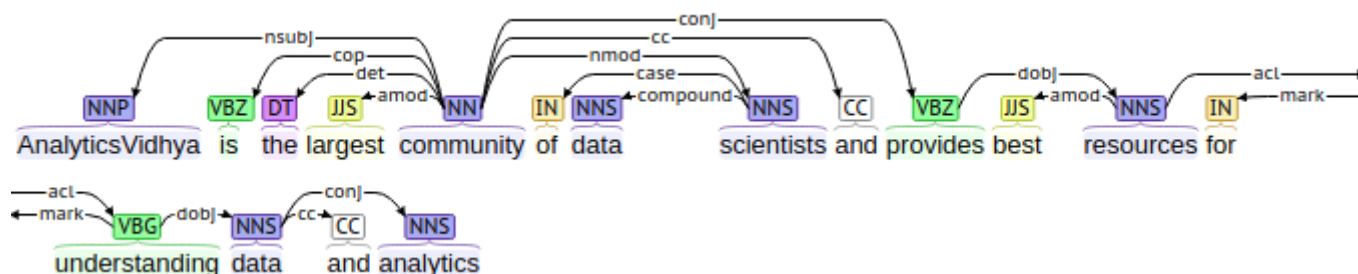
Download Brochure

Structural aspects of the text refer to the organization of tokens in a sentence and the how the contexts among them are interrelated. This organization is often depicted by the word-to-word grammar relationships which are also known as dependencies. Dependency is the notion that syntactic units (words) are connected to each other by directed links which describe the relationship possessed by the connected words.

Clausal Argument Relations	Description
NSUBJ	Nominal subject
DOBJ	Direct object
IOBJ	Indirect object
CCOMP	Clausal complement
XCOMP	Open clausal complement
Nominal Modifier Relations	Description
NMOD	Nominal modifier
AMOD	Adjectival modifier
NUMMOD	Numeric modifier
APPOS	Appositional modifier
DET	Determiner
CASE	Prepositions, postpositions and other case markers
Other Notable Relations	Description
CONJ	Conjunct
CC	Coordinating conjunction

These dependencies map directly onto a directed graph representation, in which words in the sentence are nodes in the graph and grammatical relations are edge labels. This directed graph representation is also called as the dependency tree. For example, the dependency tree of the sentence is shown in the figure below:

AnalyticsVidhya is the largest community of data scientists and provides best resources for understanding data and analytics.



Another way to represent this tree is following:

- > community-NN (**root**)
- > AnalyticsVidhya-NNP (**nsubj**)
- > is-VBZ (**cop**)
- > the-DT (**det**)
- > largest-JJS (**amod**)
- > scientists-NNS (**pobj**)

Subscribe!

Download Brochure

-> *of-IN* (**prep**)
 -> *data-NNS* (**case**)
 -> *and-CC* (**cc**)
 -> *provides-VBZ* (**conj**)
 -> *resources-NNS* (**dobj**)
 -> *best-JJS* (**amod**)
 -> *understanding-VBG* (**pcomp**)
 -> *for-IN* (**mark**)
 -> *data-NNS* (**dobj**)
 -> *and-CC* (**cc**)
 -> *analytics-NNS* (**conj**)

1. In this graphical representation of sentence, each term is represented in the pattern of “ -> Element_A – Element_B (Element_C) “.
2. Element_A represents the word, Element_B represents the part of speech tag of word, Element C represents the grammar relation among the word and its parent node, and the indentation before the symbol “ -> “ represents the level of a word in the tree. [Here](https://nlp.stanford.edu/software/dependencies_manual.pdf) (https://nlp.stanford.edu/software/dependencies_manual.pdf) is the reference list to understand the dependency relations.
3. The tree shows that the term “community” is the structural centre of the sentence and is represented as root linked by 7 nodes (“AnalyticsVidhya“, “is“, “the“, “largest“, “and“, “scientists“, “provides“).
4. Out of these 7 connected nodes, the terms “scientists” and “provides” are the root nodes of two other subtrees.
5. Each subtree is a itself a dependency tree with relations such as – (“provides” <-> “resources” <by> “dobj” relation), (“resources” <-> “best” <by> “amod” relation).

These trees can be generated in python using libraries such as NLTK, Spacy or Stanford-CoreNLP and can be used to obtain subject-verb-object triplets, noun and verb phrases, grammar dependency relationships, and part of speech tags etc for example –

-> scientists-NNS (pobj)	Grammar: <prep> <nn> <pobj>
-> of-IN (prep)	POS: IN – NNS – NNS
-> data-NNS (nn)	Phrase: of data scientist

[Subscribe!](#)
[Download Brochure](#)

Grammar: <dobj> <cc> <conj>

-> understanding-VBG (pcomp)

POS: NNS – CC – NNS

-> for-IN (prep)

Phrase: data and analytics

-> data-NNS (dobj)

-> and-CC (cc)

Grammar: <prep> <pcomp> <prep> <dobj>

-> analytics-NNS (conj)

POS: VBG – IN – NNS

Phrase: for understanding data and analytics

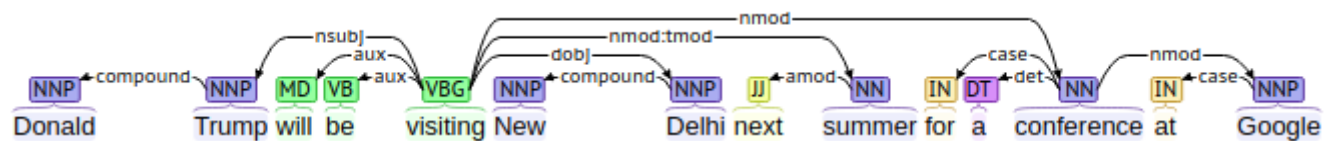
Applications of Dependency Trees

Named Entity Recognition

Named-entity recognition (NER) is the process of locating and classifying named entities in a textual data into predefined categories such as the names of persons, organizations, locations, expressions of times, quantities, monetary values, percentages, etc.

To recognize the named entities, one needs to parse the dependency tree in a top to bottom manner and identify the noun phrases. Noun phrases are the subtrees which are connected by a relation from nouns family such as nsubj, nobj, dobj etc nmod etc. For example –

Donald Trump will be visiting New Delhi next summer for a conference at Google



-> visiting-VBG (root)

-> Trump-NNP (nsubj)

-> Donald-NNP (compound)

-> will-MD (aux)

-> be-VB (aux)

-> Delhi-NNP (dobj)

Subscribe!

Download Brochure

-> *New*-NNP (**compound**)
-> *summer*-NN (**tmod**)
-> *next*-JJ (**amod**)
-> *conference*-NN (**nmod**)
-> *for*-IN (**case**)
-> *a*-DT (**det**)
-> *Google*-NNP (**nmod**)
-> *at*-IN (**case**)

In the above trees, following noun phrases are detected from the grammar relations of noun family such as:

Trump <-> visiting <by> nsubj
Delhi <-> visiting <by> dobj
summer <-> Delhi <by> nmod
conference <-> visiting <by> nmod
Google <-> conference <by> nmod

Named entities can be obtained by identifying the NNP (proper noun) part of speech tag of the root node. Example – Trump, Delhi, and Google have the part of speech tag NNP. To generate the proper noun phrase linked with root node, one needs to parse the subtree linked with the root nodes. Using the grammar rules, following named entities are obtained:

<compound> <nsubj> : Donald Trump
<compound> <dsubj> : New Delhi
<nmod> : Google

Other tasks such as phrase chunking and entity wise sentiment analysis can be performed using similar processes. For example, one sentence may contain multiple sentiments, contexts and entities and dictionary based models may not perform well. In the following sentence, there are two contexts:

Sentence – His acting was good but the script was poor

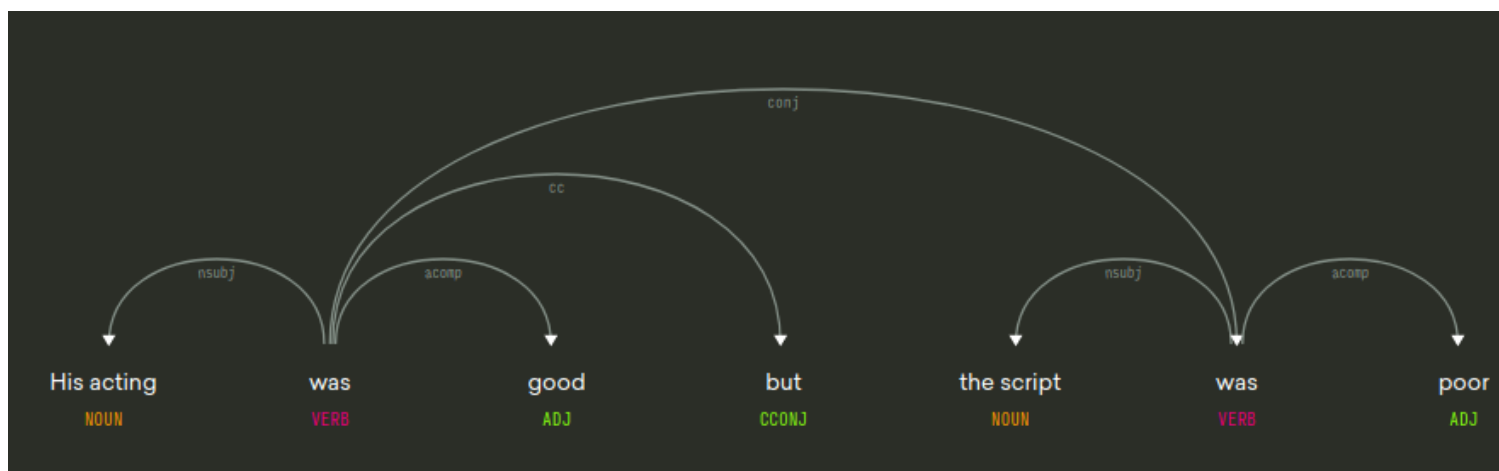
Context 1: His acting was good

Context 2: the script was poor

Both of the contexts have different sentiments, one is negative while other is positive. In the dependency tree of this sentence, there are two sub-trees which correspond to different contexts and can be used to extract them. The dependency tree of the sentence is

Subscribe!

Download Brochure



The subtrees can be extracted and evaluated individually from this dependency tree to compute the dependency tree. This [paper](https://nlp.stanford.edu/pubs/joint-parse-ner.pdf) (https://nlp.stanford.edu/pubs/joint-parse-ner.pdf) from Stanford and this [paper](https://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14741/14133) (https://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14741/14133) from Singapore University describes the efficient approaches to perform NER using dependencies.

Coreference Resolution or Anaphora Resolution

Coreference resolution is the task of finding all expressions that refer to the same entity in a text. It is an important step for a lot of higher level NLP tasks that involve natural language understanding such as document summarization, question answering, and information extraction.

"I voted for Nader because he was most aligned with my values," she said.

The coreference process

1. The first step is to create dependency tree of the text. Once a dependency tree is created, it needs to be traversed recursively to create multiple features such as:

John telephoned Bill. He lost his laptop.

Subscribe!

Download Brochure

```
-> telephoned-VBD (root)
-> John-NNP (nsubj)
-> Bill-NNP (dobj)
```

```
-> lost-VBD (root)
-> He-PRP (nsubj)
-> laptop-NN (dobj)
-> the-DT (det)
```

Np -> noun phrases nodes

Hw -> headwords

RI -> grammar relations

Lv -> level in the tree

Pos -> part of speech tag

Gen -> gender of part of speech tag

2. Identify the tokens having the part of speech tag of pronoun family, (identified by PRP tag) and the Proper Nouns or Named entities. By going through different uses of the pronouns. They can be classified in different groups according to the usage.

Type	Pronouns
Male Pronouns	["he", "him", "himself", "his"]
Female Pronouns	["her", "hers", "herself", "she"]
Plural Pronouns	["we", "us", "ourselves", "ours", "our", "they", "them", "themselves", "theirs", "their"]
Singular Pronouns	["i", "me", "myself", "mine", "my", "yourself"]
Neutral Pronouns1	["it", "its", "itself"]
Neutral Pronouns2	["here", "there"]
Other Pronouns	["who", "whom", "whose"]

Using Named Entity Recognizer, identified named entities are Bill, John

Using gender api such as this, gender of named entities are Male entities.

Subscribe!

Download Brochure

Features of sentences:

```

0
sent  John telephoned Bill.
np    ['John', 'Bill']
np,hw [('John', 'John'), ('Bill', 'Bill')]
np,rl  [('John', 'nsubj'), ('Bill', 'dobj')]
np,lv  [('John', 1), ('Bill', 1)]
np,pos  [('John', 'NNP'), ('Bill', 'NNP')]
np,gen [(u'male', 0), (u'male', 0)]

```

```

1
sent  He lost the laptop.
np    ['He', 'the laptop']
np,hw [('He', 'He'), ('the laptop', 'laptop')]
np,rl  [('He', 'nsubj'), ('the laptop', 'dobj')]
np,lv  [('He', 1), ('the laptop', 1)]
np,pos  [('He', 'PRP'), ('the laptop', 'NN')]
np,gen [(u'male', 0), (u'male', 0)]

```

3. Map the pronoun tokens with named entity tokens. Start from the bottommost sentence features and identify part of speech tag and its gender. To properly map the tokens, the properties/features are exploited.

3.1 map the tokens with same gender of pronoun and named entity

3.2 map the tokens with same singularity / plurality

3.3 map the tokens with same grammar relations

This [paper \(http://www.aclweb.org/anthology/C10-1068\)](http://www.aclweb.org/anthology/C10-1068) from Soochow University describes the use of dependency trees in coreference resolution.

Question Answering

Another important task in which computational linguistics can help to obtain results with high relevance is the Question Answering which is treated as one of the hardest tasks involved with text data. Question answering systems based on computational linguistics uses the syntactic structures of the query questions and matches them with the responses having similar syntactic structures. The similar syntactic structures contribute the answer set to a particular question. For example

Question: What is the capital of India?

Subscribe!

Download Brochure

Answer: New Delhi is the capital of India

-> capital-NN (root)
-> what-WP (nsubj)
-> is-VBZ (cop)
-> the-DT (det)
-> of-IN (prep)
-> India-NNP (pobj)

-> capital-NN (root)
-> Delhi-NNP (nsubj)
-> New-NNP (nn)
-> is-VBZ (cop)
-> the-DT (det)
-> of-IN (prep)
-> India-NNP (pobj)

Both the question and answer dependency trees have similar patterns and can be used to generate the answer responses to specific queries. This [paper1](http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.1.4864&rep=rep1&type=pdf) (<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.1.4864&rep=rep1&type=pdf>) and [paper2](http://cogcomp.org/papers/PunayakanokRoYi04.pdf) (<http://cogcomp.org/papers/PunayakanokRoYi04.pdf>) describe the approaches to perform question answering using dependency trees.

Other Tasks that uses Computational Linguistics

- Machine Translation
- Text Summarization
- Natural Language Generation
- Natural Language Understanding
- Speech to Text

Generating Dependency Trees using Stanford Core NLP

Subscribe!

Download Brochure

```
from stanfordcorenlp import StanfordCoreNLP as scn
nlp = scn(r'/path/to/stanford-corenlp-full-2017-06-09/')

sentence = 'His acting was good but script was poor'

print
print 'Part of Speech:'
print nlp.pos_tag(sentence)

print
print 'Dependency Parsing:'
print nlp.dependency_parse(sentence)

[(u'His', u'PRP$'),
 (u'acting', u'NN'),
 (u'was', u'VBD'),
 (u'good', u'JJ'),
 (u'but', u'CC'),
 (u'script', u'NN'),
 (u'was', u'VBD'),
 (u'poor', u'JJ')]

[(u'ROOT', 0, 4),
 (u'nmod:poss', 2, 1),
 (u'nsubj', 4, 2),
 (u'cop', 4, 3),
 (u'cc', 4, 5),
 (u'nsubj', 8, 6),
 (u'cop', 8, 7),
 (u'conj', 4, 8)]
```

Dependency Trees using Spacy

[Subscribe!](#)[Download Brochure](#)

```
import spacy

nlp = spacy.load("en")

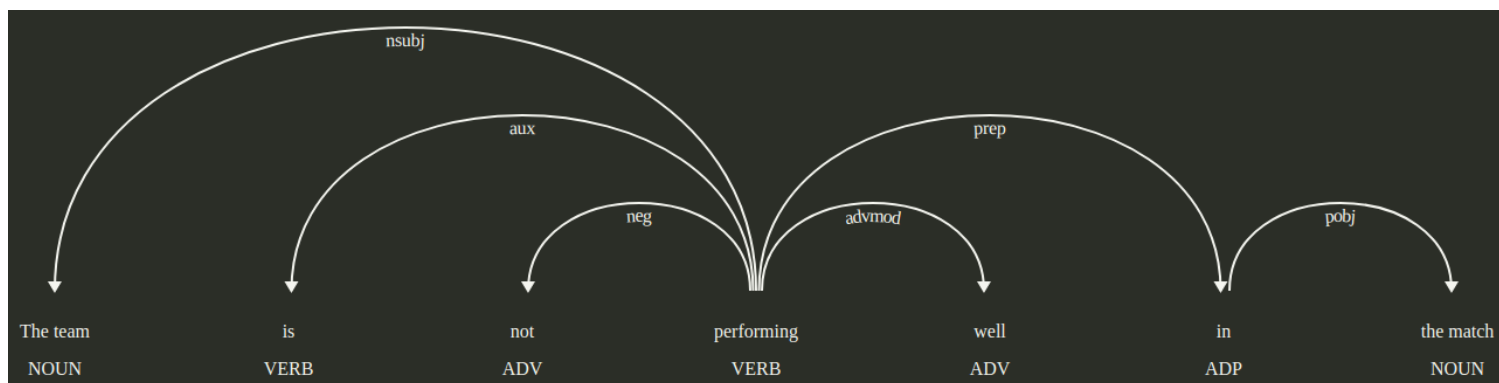
doc = nlp(u'The team is not performing well in the match')

for token in doc:
    print str(token.text), str(token.lemma_), str(token.pos_), str(token.dep_)

>>> The, the, NOUN, nsubj
>>> team, team, Noun, nsubj
>>> Is, is, VERB, aux
>>> Not, not, ADV, neg
>>> Performing, perform, VERB, root
>>> Well, well, ADV, advmod
>>> In, in, ADP, prep
>>> The, the, Noun, pobj
>>> Match, match ,Noun, pobj

from spacy import displacy

displacy.serve(doc, style='dep')
```

[Subscribe!](#)[Download Brochure](#)

Note: If you wish to explore this field further, then have a look at our detailed [video course on NLP](https://trainings.analyticsvidhya.com/courses/course-v1:AnalyticsVidhya+NLP101+2018_T1/about?utm_source=blog&utm_medium=introduction-computational-linguistics-dependency-trees) (https://trainings.analyticsvidhya.com/courses/course-v1:AnalyticsVidhya+NLP101+2018_T1/about?utm_source=blog&utm_medium=introduction-computational-linguistics-dependency-trees).

End Notes

In this article, I discussed the field of computational linguistics and how grammar relations among the sentences can be used in different tasks related to text data. If you feel, there are any other resources, tasks related to dependency trees and computational linguistics that I have missed, please feel free to comment with your suggestions and feedback.







Learn (<https://www.analyticsvidhya.com/blog>), **engage** (<http://discuss.analyticsvidhya.com/>), **hack** (<https://datahack.analyticsvidhya.com/>) and **get hired** (<https://www.analyticsvidhya.com/jobs/#/user/>)!

You can also read this article on Analytics Vidhya's Android APP



(https://play.google.com/store/apps/details?id=com.analyticsvidhya.android&utm_source=blog_article&utm_campaign=blog&pcampaignid=MKT-Other-global-all-co-prtnr-py-PartBadge-Mar2515-1).

Share this:

-  (<https://www.analyticsvidhya.com/blog/2017/12/introduction-computational-linguistics-dependency-trees/?share=linkedin&nb=1>)
-  (<https://www.analyticsvidhya.com/blog/2017/12/introduction-computational-linguistics-dependency-trees/?share=facebook&nb=1>)
-  (<https://www.analyticsvidhya.com/blog/2017/12/introduction-computational-linguistics-dependency-trees/?share=google-plus-1&nb=1>)
-  (<https://www.analyticsvidhya.com/blog/2017/12/introduction-computational-linguistics-dependency-trees/?share=twitter&nb=1>)
-  (<https://www.analyticsvidhya.com/blog/2017/12/introduction-computational-linguistics-dependency-trees/?share=pocket&nb=1>)
-  (<https://www.analyticsvidhya.com/blog/2017/12/introduction-computational-linguistics-dependency-trees/?share=reddit&nb=1>)

Like this:

Like

Be the first to like this.

Subscribe!

Download Brochure



TAGS : [COMPUTATIONAL LINGUISTICS \(HTTPS://WWW.ANALYTICSVIDHYA.COM/BLOG/TAG/COMPUTATIONAL-LINGUISTICS/\)](https://www.analyticsvidhya.com/blog/tag/computational-linguistics/),
[COREFERENCE RESOLUTION \(HTTPS://WWW.ANALYTICSVIDHYA.COM/BLOG/TAG/COREFERENCE-RESOLUTION/\)](https://www.analyticsvidhya.com/blog/tag/coreference-resolution/), [DEPENDENCY TREES \(HTTPS://WWW.ANALYTICSVIDHYA.COM/BLOG/TAG/DEPENDENCY-TREES/\)](https://www.analyticsvidhya.com/blog/tag/dependency-trees/), [NAMED ENTITY RECOGNITION \(HTTPS://WWW.ANALYTICSVIDHYA.COM/BLOG/TAG/NAMED-ENTITY-RECOGNITION/\)](https://www.analyticsvidhya.com/blog/tag/named-entity-recognition/), [NATURAL LANGUAGE PROCESSING \(HTTPS://WWW.ANALYTICSVIDHYA.COM/BLOG/TAG/NATURAL-LANGUAGE-PROCESSING/\)](https://www.analyticsvidhya.com/blog/tag/natural-language-processing/), [NLP \(HTTPS://WWW.ANALYTICSVIDHYA.COM/BLOG/TAG/NLP/\)](https://www.analyticsvidhya.com/blog/tag/nlp/), [QUESTION ANSWERING \(HTTPS://WWW.ANALYTICSVIDHYA.COM/BLOG/TAG/QUESTION-ANSWERING/\)](https://www.analyticsvidhya.com/blog/tag/question-answering/), [SPACY \(HTTPS://WWW.ANALYTICSVIDHYA.COM/BLOG/TAG/SPACY/\)](https://www.analyticsvidhya.com/blog/tag/spacy/), [STANFORD CORE NLP \(HTTPS://WWW.ANALYTICSVIDHYA.COM/BLOG/TAG/STANFORD-CORE-NLP/\)](https://www.analyticsvidhya.com/blog/tag/stanford-core-nlp/)

NEXT ARTICLE

15 Trending Data Science GitHub Repositories you can not miss in 2017

(<https://www.analyticsvidhya.com/blog/2017/12/15-data-science-repositories-github-2017/>)

...

PREVIOUS ARTICLE

Essentials of Deep Learning : Introduction to Long Short Term Memory

(<https://www.analyticsvidhya.com/blog/2017/12/fundamentals-of-deep-learning-introduction-to- lstm/>)



(<https://www.analyticsvidhya.com/blog/author/shivam5992/>)

Shivam Bansal

(<https://www.analyticsvidhya.com/blog/author/shivam5992/>)

Shivam Bansal is a data scientist with exhaustive experience in Natural Language Processing and Machine Learning in several domains. He is passionate about learning and always looks forward to solving

[Subscribe](#)

[Download Brochure](#)

challenging analytical problems.

[_](https://twitter.com/shivamshaz) (https://twitter.com/shivamshaz) [in](https://www.linkedin.com/in/shivambansal1) (https://www.linkedin.com/in/shivambansal1)

[_](https://github.com/shivam5992) (https://github.com/shivam5992).

RELATED ARTICLES

[SHUVAYAN DAS \(HTTPS://WWW.ANALYTICUNAL JAIN \(HTTPS://WWW.ANALYTICS](https://www.analyticsvidhya.com/blog/2017/12/introduction-computational-linguistics-dependency-trees/) [ANALYTICS VIDHYA CONTENT TEAM \(HT...](https://www.analyticsvidhya.com/blog/2017/12/introduction-computational-linguistics-dependency-trees/)



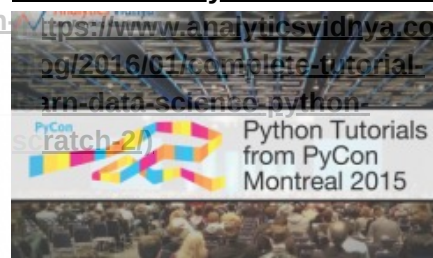
(https://www.analyticsvidhya.com/blog/2017/12/introduction-computational-linguistics-dependency-trees/)
[r-python-nosql-manager/](https://www.analyticsvidhya.com/blog/2017/12/introduction-computational-linguistics-dependency-trees/)) [tutorial-learn-data-science-python-scratch-2/](https://www.analyticsvidhya.com/blog/2017/12/introduction-computational-linguistics-dependency-trees/)) [description-megastar/](https://www.analyticsvidhya.com/blog/2017/12/introduction-computational-linguistics-dependency-trees/))

Getting Mongo-ed in NoSQL manager, R & Python

[FAIZAN SHAIKH \(HTTPS://WWW.ANALYTICUNAL JAIN \(HTTPS://WWW.ANALYTICS](https://www.analyticsvidhya.com/blog/2017/12/introduction-computational-linguistics-dependency-trees/)
[blog/2017/12/introduction-computational-linguistics-dependency-trees/](https://www.analyticsvidhya.com/blog/2017/12/introduction-computational-linguistics-dependency-trees/)
[nosql-manager/](https://www.analyticsvidhya.com/blog/2017/12/introduction-computational-linguistics-dependency-trees/)



A Complete Tutorial to Learn Data Science with Python from Scratch



Hackathon Problem Description: Do you know who's a Megastar?

[QUEST BLOG \(HTTPS://WWW.ANALYTICS](https://www.analyticsvidhya.com/blog/2017/12/introduction-computational-linguistics-dependency-trees/)

(https://www.analyticsvidhya.com/blog/2017/12/introduction-computational-linguistics-dependency-trees/)
[an-image-captioning-task-using-deep-learning/](https://www.analyticsvidhya.com/blog/2017/12/introduction-computational-linguistics-dependency-trees/)) [montreal-2015-data-science-workshops/](https://www.analyticsvidhya.com/blog/2017/12/introduction-computational-linguistics-dependency-trees/)) [learning-models-as-apis-using-flask/](https://www.analyticsvidhya.com/blog/2017/12/introduction-computational-linguistics-dependency-trees/))

Automatic Image Captioning using Deep Learning (CNN and LSTM) in PyTorch

[\(https://www.analyticsvidhya.com/blog/2018/04/solving-an-image-captioning-task-using-deep-learning/\)](https://www.analyticsvidhya.com/blog/2018/04/solving-an-image-captioning-task-using-deep-learning/)

PyCon Montreal 2015 tutorials – Hands-on way to learn Data Science in Python

[\(https://www.analyticsvidhya.com/blog/2015/04/pycon-montreal-2015-data-science-workshops/\)](https://www.analyticsvidhya.com/blog/2015/04/pycon-montreal-2015-data-science-workshops/)

Tutorial to deploy Machine Learning models in Production as APIs (using Flask)

[\(https://www.analyticsvidhya.com/blog/2017/09/machine-learning-models-as-apis-using-flask/\)](https://www.analyticsvidhya.com/blog/2017/09/machine-learning-models-as-apis-using-flask/)

This article is quite old and you might not get a prompt response from the author. We request you to post this comment on Analytics Vidhya's [discussion portal \(https://discuss.analyticsvidhya.com/\)](https://discuss.analyticsvidhya.com/) to get your queries resolved

8 COMMENTS

Subscribe!

Download Brochure

**KRISHNASAMY SELVAN**[Reply](#)

December 14, 2017 at 10:32 am (<https://www.analyticsvidhya.com/blog/2017/12/introduction-computational-linguistics-dependency-trees/#comment-147585>).

Appreciate if anyone can provide a case study/code on Q-learning technique. Very Urgent.. Thanks in advance..
91-9900086995

**DAVID LEE**[Reply](#)

December 14, 2017 at 11:17 am (<https://www.analyticsvidhya.com/blog/2017/12/introduction-computational-linguistics-dependency-trees/#comment-147588>).

Very neat and clean explanation of structural aspects in text data. Thanks for sharing the article on Computational Linguistics.

**ASHWIN PERTI (HTTP://ABES.AC.IN)**[Reply](#)

December 14, 2017 at 11:17 am (<https://www.analyticsvidhya.com/blog/2017/12/introduction-computational-linguistics-dependency-trees/#comment-147589>).

Well explained NLP

**AMIT J**[Reply](#)

December 14, 2017 at 2:49 pm (<https://www.analyticsvidhya.com/blog/2017/12/introduction-computational-linguistics-dependency-trees/#comment-147611>).

How can I install Spacy in Windows.

**LINGAM JETTA**[Reply](#)

December 19, 2017 at 8:52 pm (<https://www.analyticsvidhya.com/blog/2017/12/introduction-computational-linguistics-dependency-trees/#comment-148709>).

-> pip install spacy
after this download the whl file required for spacy
i.e. spacy, cymem, preshed, murmurhash, ujson, thinc
and run this command.

-> pip install spacy-2.0.5-cp27-cp27m-win_amd64.whl cymem-1.31.2-cp27-cp27m-win_amd64.whl preshed-1.0.0-cp27-cp27m-win_amd64.whl murmurhash-0.28.0-cp27-cp27m-win_amd64.whl thinc-6.10.2-cp27-cp27m-win_amd64.whl ujson-1.35-cp27-cp27m-win_amd64.whl

[Subscribe!](#)[Download Brochure](#)

**MANOHAR RAVI**[Reply](#)

December 15, 2017 at 6:15 pm (<https://www.analyticsvidhya.com/blog/2017/12/introduction-computational-linguistics-dependency-trees/#comment-147850>)

Great and useful info

**KRISHNA**[Reply](#)

December 19, 2017 at 6:53 pm (<https://www.analyticsvidhya.com/blog/2017/12/introduction-computational-linguistics-dependency-trees/#comment-148692>)

Hi i have two queations –

Which library do you prefer for good accuracy of dependency trees?

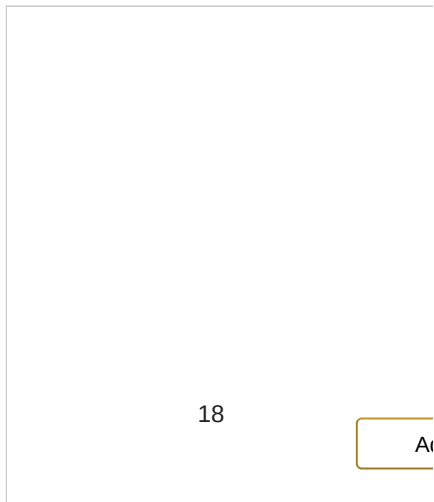
How can you use these trees in text classification.

**MILA P**[Reply](#)






February 7, 2018 at 5:20 pm (<https://www.analyticsvidhya.com/blog/2017/12/introduction-computational-linguistics-dependency-trees/#comment-151224>)

Nice Info

[Subscribe!](#)[Download Brochure](#)



TOP ANALYTICS VIDHYA USERS

Rank	Name		Points
1		SRK (https://datahack.analyticsvidhya.com/user/profile/SRK)	9688
2		vopani (https://datahack.analyticsvidhya.com/user/profile/vopani)	9200
3		aayushmnit (https://datahack.analyticsvidhya.com/user/profile/aayushmnit)	7779
4		mark12 (https://datahack.analyticsvidhya.com/user/profile/mark12)	7212
5		sonny (https://datahack.analyticsvidhya.com/user/profile/sonny)	5947

[More Rankings \(http://datahack.analyticsvidhya.com/users\)](http://datahack.analyticsvidhya.com/users)

Subscribe!

Download Brochure



POPULAR POSTS

24 Ultimate Data Science Projects To Boost Your Knowledge and Skills (& can be accessed freely)
(<https://www.analyticsvidhya.com/blog/2018/05/24-ultimate-data-science-projects-to-boost-your-knowledge-and-skills/>)

A Complete Tutorial to Learn Data Science with Python from Scratch
(<https://www.analyticsvidhya.com/blog/2016/01/complete-tutorial-learn-data-science-python-scratch-2/>)

The 25 Best Data Science and Machine Learning GitHub Repositories from 2018
(<https://www.analyticsvidhya.com/blog/2018/12/best-data-science-machine-learning-projects-github/>)

Subscribe!

Download Brochure

Essentials of Machine Learning Algorithms (with Python and R Codes)

(<https://www.analyticsvidhya.com/blog/2017/09/common-machine-learning-algorithms/>)

Understanding Support Vector Machine algorithm from examples (along with code)

(<https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/>)

7 Types of Regression Techniques you should know!

(<https://www.analyticsvidhya.com/blog/2015/08/comprehensive-guide-regression/>)

A Technical Overview of AI & ML (NLP, Computer Vision, Reinforcement Learning) in 2018 & Trends for 2019

(<https://www.analyticsvidhya.com/blog/2018/12/key-breakthroughs-ai-ml-2018-trends-2019/>)

6 Easy Steps to Learn Naive Bayes Algorithm (with codes in Python and R)

(<https://www.analyticsvidhya.com/blog/2017/09/naive-bayes-explained/>)



RECENT POSTS

Build your First Image Classification Model in just 10 Minutes!

(<https://www.analyticsvidhya.com/blog/2019/01/build-image-classification-model-10-minutes/>)

JANUARY 10, 2019

A Hands-On Introduction to Time Series Classification (with Python Code)

(<https://www.analyticsvidhya.com/blog/2019/01/introduction-time-series-classification/>)

JANUARY 7, 2019

DataHack Radio #15: Exploring the Applications & Potential of Reinforcement Learning with Xander Steenbrugge (<https://www.analyticsvidhya.com/blog/2019/01/datahack-radio-reinforcement-learning-xander/>)

Subscribe!

Download Brochure

JANUARY 6, 2019

A Comprehensive Learning Path for Deep Learning in 2019**(<https://www.analyticsvidhya.com/blog/2019/01/comprehensive-learning-path-deep-learning-2019/>)**

JANUARY 2, 2019

[http://www.edvancer.in/certified-data-scientist-with-python-course?](http://www.edvancer.in/certified-data-scientist-with-python-course?utm_source=AV&utm_medium=AVads&utm_campaign=AVadsnonfc&utm_content=pythonavad)[utm_source=AV&utm_medium=AVads&utm_campaign=AVadsnonfc&utm_content=pythonavad](http://www.edvancer.in/certified-data-scientist-with-python-course?utm_source=AV&utm_medium=AVads&utm_campaign=AVadsnonfc&utm_content=pythonavad)[https://trainings.analyticsvidhya.com/courses/course-](https://trainings.analyticsvidhya.com/courses/course-v1:AnalyticsVidhya+DS101+2018T2/about?utm_source=Blog&utm_medium=Sticky_banner1)[v1:AnalyticsVidhya+DS101+2018T2/about?utm_source=Blog&utm_medium=Sticky_banner1](https://trainings.analyticsvidhya.com/courses/course-v1:AnalyticsVidhya+DS101+2018T2/about?utm_source=Blog&utm_medium=Sticky_banner1)[Subscribe!](#)[Download Brochure](#)



Analytics Vidhya
Learn everything about analytics



Analytics Vidhya
Learn everything about analytics

ANALYTICS VIDHYA

About Us
(<http://www.analyticsvidhya.com/about-me/>)

Our Team
(<https://www.analyticsvidhya.com/about-me/team/>)

Career
(<https://www.analyticsvidhya.com/career-analytics-vidhya/>)

Contact Us
(<https://www.analyticsvidhya.com/contact>)

Write for us
(<https://www.analyticsvidhya.com/about-me/write/>)

DATA SCIENTISTS

Blog
(<https://www.analyticsvidhya.com/blog>)

Hackathon
(<https://datahack.analyticsvidhya.com/>)

Discussions
(<https://discuss.analyticsvidhya.com/>)

Apply Jobs
(<https://www.analyticsvidhya.com/career-jobs>)

Leaderboard
(<https://datahack.analyticsvidhya.com/cv10percent/>)

COMPANIES

Post Jobs
(<https://www.analyticsvidhya.com/corporate/>)

Trainings
(<https://trainings.analyticsvidhya.com/>)

Hiring
(<https://datahack.analyticsvidhya.com/>)

Advertising
(<https://www.analyticsvidhya.com/contact>)

Reach Us
(<https://www.analyticsvidhya.com/contact>)

JOIN OUR COMMUNITY :

f <https://www.facebook.com/analyticsvidhya/> 19252 Followers
(<https://plus.google.com/+Analyticsvidhya>)

t <https://twitter.com/analyticsvidhya> 48336 Followers
(<https://plus.google.com/+Analyticsvidhya>)

in <https://www.linkedin.com/company/analyticsvidhya/> 3054 Followers
(<https://plus.google.com/+Analyticsvidhya>)

G+ <https://plus.google.com/+Analyticsvidhya> 7513 Followers
(<https://plus.google.com/+Analyticsvidhya>)

v1:AnalyticsVidhya+CVDL101+CVDL101_T1/about?utm_source=Blog&utm_medium=Sticky_banner&utm_campaign=CV10percent

Subscribe!

Download Brochure