

NLP Project Report

Sentence Simplification

Freya Mehta (20171184) & Aamir Farhan (20161078)

25th November, 2019

Contents

1. Project Description
2. Introduction
3. Literature Review
4. Procedure
5. Results
6. Error Analysis
7. Caveats
8. Paper Review
9. References

Project Description

This project aims to identify english sentences with more than two clauses and mark the clause boundaries.

Additionally, breaking the complex/compound sentences into multiple simple sentences.

We have incorporated non-destructive simplification. It is a type of process which reduces the average sentence length and complexity to make text simpler. This syntactic text simplification focuses on extracting embedded clauses from structurally complex sentences.

Following a rule-based method for this purpose, we primarily use constituency-based parse tree.

Introduction

Sentence Clause Structure:


A **clause** is a group of words that functions as one part of speech and that includes a verb phrase. A simple sentence consists of only one clause. A complex sentence has at least one independent clause plus at least one [dependent clause](#).

Coordinate Clause (Main clause) : Coordination joins two independent clauses that contain related ideas of equal importance.

Original sentences: I spent my entire paycheck last week. I am staying home this weekend.

Revised sentence: I spent my entire paycheck last week, so I am staying home this weekend.

Subordinate Clause (Relative Clause) : Subordination joins two sentences with related ideas by merging them into a main clause (a complete sentence) and a dependent



clause (a construction that relies on the main clause to complete its meaning).

Original sentences: Tracy stopped to help the injured man.
She would be late for work.

Revised sentence: Even though Tracy would be late for work, she stopped to help the injured man.

Rule based approach : By identifying the relations between the main verb and coordinate/subordinate verbs, we can easily classify the clauses, identify the possible relations (Rules) between the verbs in the clauses and determine the most optimal clause boundary in that sentence based on word order.

Rule based approach using **constituency parse tree** is efficient over dependency tree because constituency-parse is based on phrase structure grammar, which is the most relevant if one is seeking to extract clauses from a sentence. It can be done using dependencies as well, but in that case, one will essentially be reconstructing the phrase structure -- starting from the root and looking at dependent nodes.

Literature Review

The review of literature in this project is divided into two parts : (i) Phrase Structure Trees and Constituency Parsing, (ii) Clause boundary detection.

Our approach uses constituency parsing using Stanford parser to obtain the Phrase Structure Trees of Complex sentences. The clauses are extracted from these trees using the rules as described in the procedure.

The problem of computing complex sentences in natural language processing is to make sentences simple to understand, by identifying clause boundaries. [1] provides a survey of predicting clause boundaries while [2] is a rule based method for clause boundary detection. The latter method is a pipeline that uses phrase structure trees in order to determine the clauses.

Procedure

Constituency parsing is based on phrase structure grammar, which is the most relevant if you are seeking to extract clauses from a sentence.

The steps for extracting clauses are explained with the help of the following examples :

Step1

- Obtain the Phrase structure tree of the given complex sentence.

Step2

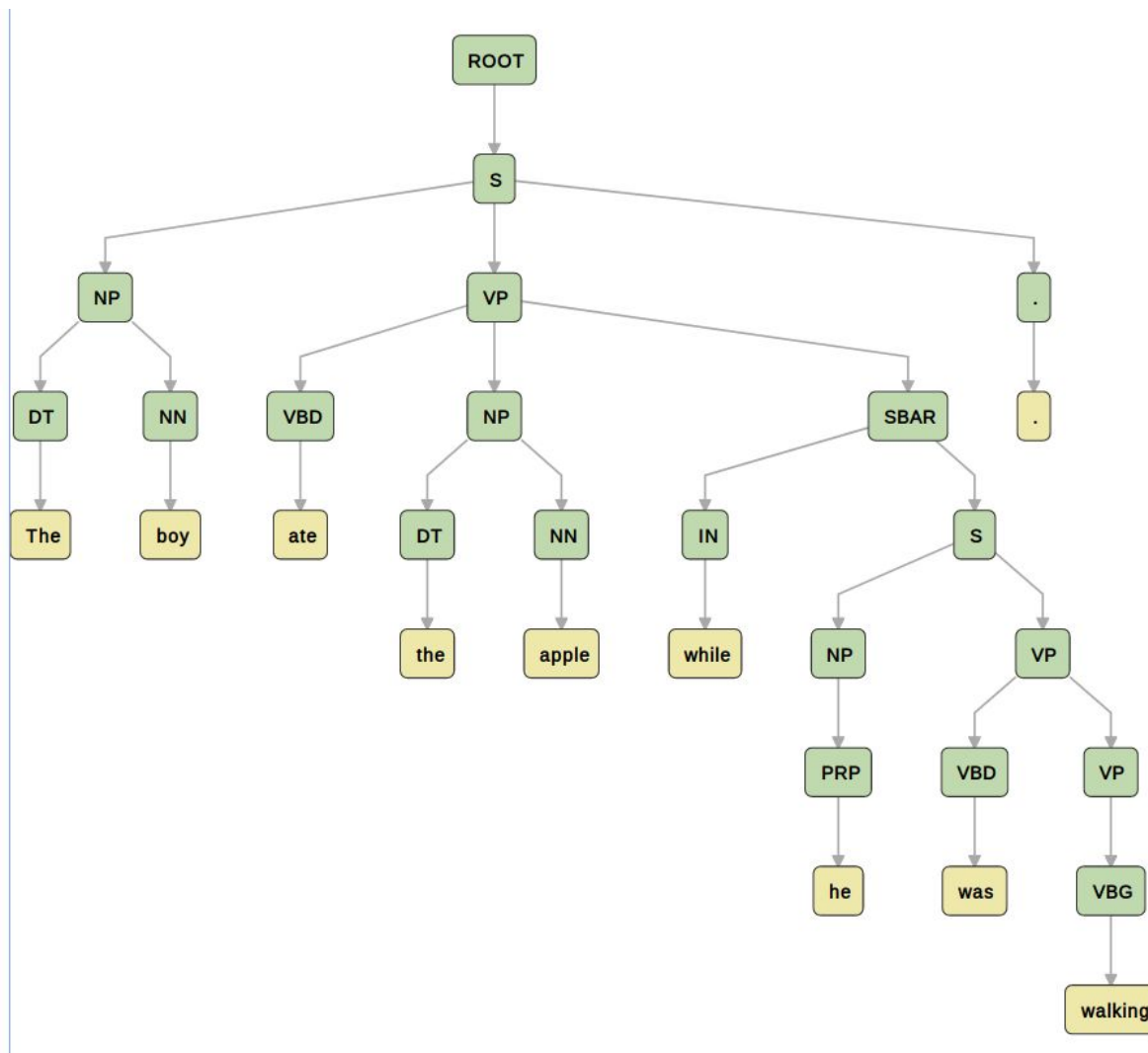
- Identify the non-root clausal nodes (SBAR, S etc) in the parse tree.
- Remove, but retain separately the subtrees rooted at these clausal nodes from the main tree.

Step3

- In the main tree (after removal of subtrees in step 2), remove any *hanging* prepositions, subordinating conjunctions and adverbs.

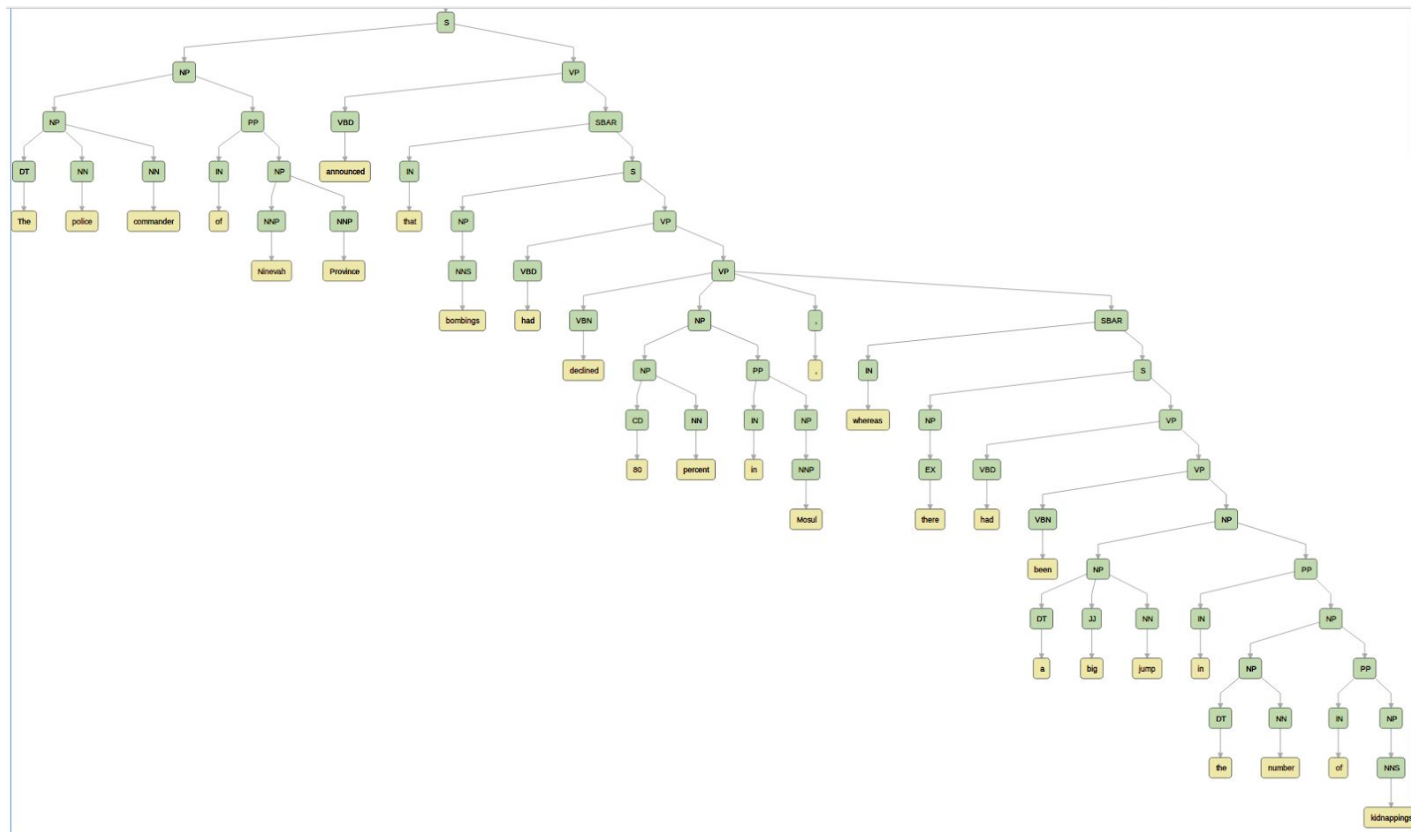
Example 1

The boy ate the apple while he was walking.



Output 1

['The boy ate the apple', 'he was walking']



Example 2

The police commander of Ninevah Province announced that bombings had declined 80 percent in Mosul, whereas there had been a big jump in the number of kidnappings.



The final clauses which this complex sentence breaks into are :

Output 2 :

['The police commander of Ninevah Province announced', 'bombings had declined 80 percent in Mosul', 'there had been a big jump in the number of kidnappings']

Results

Sentence	Clauses
The cat went to the park.	['The cat went to the park']
John likes music but does not like rock.	['John does not like rock', 'John likes music']
Joe realized that the train was late while he waited at the train station.	['Joe realized', 'the train was late', 'he waited at the train station']
The police commander of Ninevah Province announced that bombings had declined 80 percent in Mosul, whereas there had been a big jump in the number of kidnappings.	['The police commander of Ninevah Province announced', 'bombings had declined 80 percent in Mosul', 'there had been a big jump in the number of kidnappings']
Every night the office is vacuumed and dusted by the cleaning crew.	['the office is vacuumed and dusted by the cleaning crew']
For full dataset and results, click here .	

Error Analysis

A detailed error analysis revealed that the major sources of error include inaccurate sign tagging, the relatively limited coverage of the rules used to rewrite sentences, and an inability to discriminate between various subtypes of clause coordination.

For example, the complex sentence

Every night the office is vacuumed and dusted by the cleaning crew.

breaks into

['the office is vacuumed and dusted by the cleaning crew']

Because the script is not breaking the sentence at 'and'.

Caveats

- Even the best parsers (here Stanford Parser) will not always parse sentences correctly, so keep that in mind.
- Additionally, many complex sentences involve [right node raising](#), which is almost never parsed correctly by most parsers.
- You may need to modify the algorithm a little if a clause is in passive voice.

Paper Review

Iustin Dornescu, Richard Evans, Constantin Orăsan;
“Relative clause extraction for syntactic simplification”

This paper investigates syntactic simplification by an ML model developed using crfsuite, followed by a rule based method which we have incorporated in our project. This paper discusses in detail about restrictive and nonrestrictive relative clauses. Restrictive relative clauses are also called integrated, defining or identifying relative clauses. Similarly, non-restrictive relative clauses are called supplementary, appositive, non-defining or non-identifying relative clauses. It also describes the different types of noun post-modifier. This paper follows the sign complexity scheme, where punctuation marks and functional words are considered explicit markers of coordinated and subordinated constituents, the two syntactic mechanisms leading to structurally complex sentences.

Link to the paper -

<https://www.aclweb.org/anthology/W14-5601/>

References

1. Sanjeev Kumar Sharma , “Clause Boundary Identification for Different Languages: A Survey” International Journal of Computer Applications & Information Technology Vol. 8, Issue II2016(ISSN: 2278-7720)
2. Bogdan Sacaleanu, Dublin (IE); Alice Marascu, Dublin (IE); Charles Jochim, Dublin (IE) RULE-BASED SYNTACTIC APPROACH TO CLAIM BOUNDARY DETECTION IN COMPLEX SENTENCES
3. Iustin Dornescu, Richard Evans, Constantin Orășan; “Relative clause extraction for syntactic simplification”