

COMP24412 Lab 3 - The Riddle of Steel Report

Tejas Chandrasekar

April 11, 2019

0.1 Lexical Analysis

Steel is alloy of iron and carbon, and sometimes other elements. Because of its high tensile strength and low cost, it is a major component used in buildings, infrastructure, tools, ships, automobiles, machines, appliances, and weapons.

1. POS tagging the above sentence:

- NN: Noun, singular or mass
 - steel
 - alloy
 - iron
 - carbon
 - strength
 - cost
 - component
 - infrastructure
- NNS: Noun, plural
 - elements
 - buildings
 - tools
 - ships
 - automobiles
 - machines
 - appliances
 - weapons
- JJ: Adjectives
 - other
 - high
 - tensile
 - low
 - major
- RB: Adverb

- sometimes
- CC: Coordinating conjunction
 - Because
 - and
- DT: Determiner
 - an
 - a
 - its
- IN: Preposition/subordinating conjunction
 - in
 - of
- VBD: Verb, past tense
 - Used
- VZD: Verb, 3rd person singular present
 - is
- PP\$: Possessive pronoun
 - it

2. Pronominal co-references: steel, it

0.2 C-Structures

Sentence is 'Steel is an alloy of iron and carbon, and sometimes other elements.'

1. Constituency structure: see Figure 1.
2. Nominal phrasal nodes:
3. Co-ordinations:
 - Coordinating conjunction: '...iron AND carbon, AND...'
 - Gapped coordination: '...sometimes other elements.'

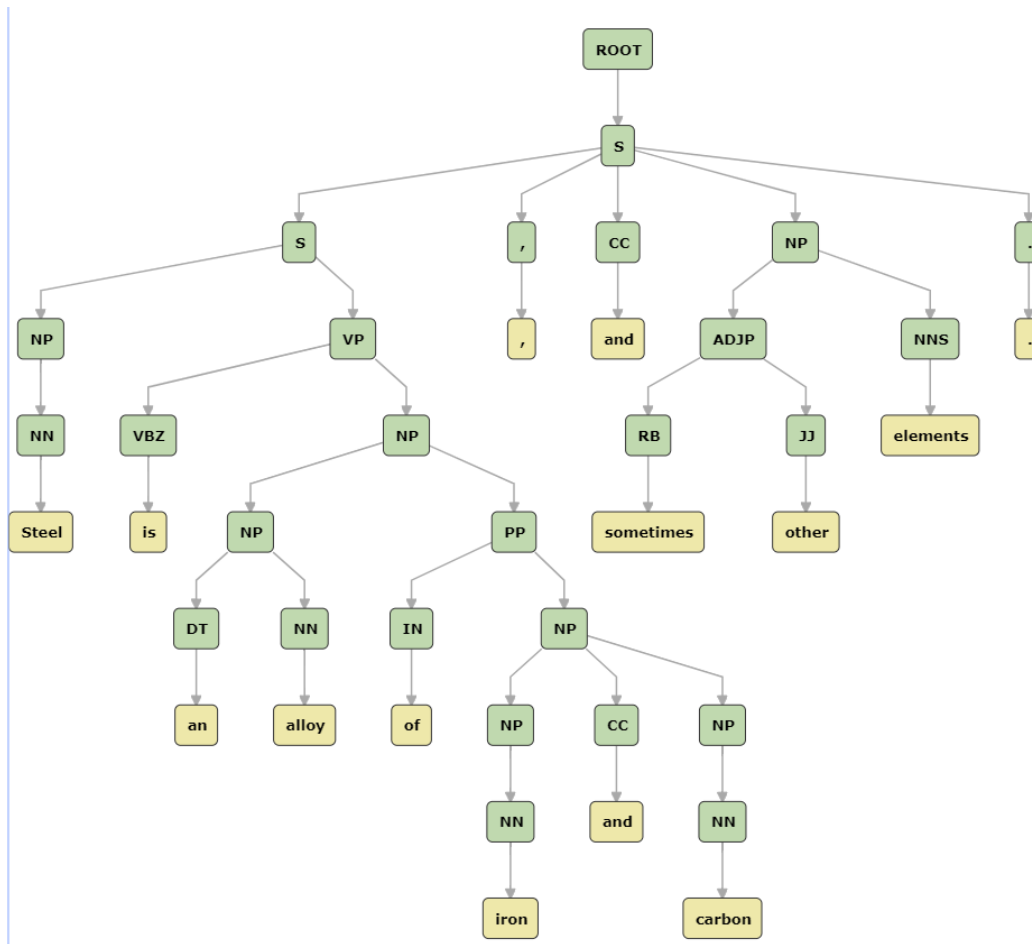


Figure 1: C-structure for the sentence

0.3 Dependencies - Exploring new territories

1. Constituency parsing divides text into sub-phrases. Using a tree structure, the types of phrases belong on branches, the individual words in the sentence are leaves, and the designation 'Sentence' is the root.

Dependency parsing connects words according to their relationships. Again using a tree structure, each node in the tree represents a word, with the leaf nodes being 'dependent' on the internal nodes, which are most often verbs.

2. Dependency structure: see Figure 2.

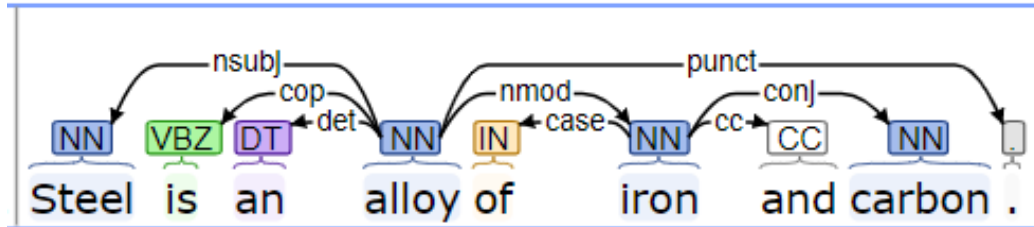


Figure 2: Dependency structure for sentence: 'Steel is an alloy of iron and carbon'

0.4 Open IE - Semantics

1. Here are the predicate-argument structures for the following sentences:
 - (a) 'Steel is an alloy.': Entity(Steel) \leftarrow Subject – relation(is) – Object \rightarrow (an alloy) Entity(alloy).
 - (b) 'Steel contains carbon.': Entity(Steel) \leftarrow Subject – relation(contains) – Object \rightarrow (carbon) Entity(carbon).
 - (c) 'Steel contains iron.': Entity(Steel) \leftarrow Subject – relation(contains) – Object \rightarrow (iron) Entity(iron).
2. Prolog translation of triples:


```
steel(X) :- alloy(X), contains(X, carbon), contains(X, iron).
```
3. RDF translation of triples:
 - (a) $\langle \text{Steel} \rangle \langle \text{isan} \rangle \langle \text{alloy} \rangle .$
 - (b) $\langle \text{Steel} \rangle \langle \text{contains} \rangle \langle \text{carbon} \rangle .$
 - (c) $\langle \text{Steel} \rangle \langle \text{contains} \rangle \langle \text{iron} \rangle .$
4. Axiom formalisation using Description Logics:

$$\text{Steel} \equiv \text{alloy} \sqcap \text{hasCarbon} \sqcap \text{hasIron}$$

5. 'Steel is an alloy of iron and carbon.' - as per the next figure, the predicate-argument structure is more complex, and the Prolog itself might differ like so:

```
steel(X) :- alloy(X, [carbon, iron]).
```

Previously, the three separate statements simply translated to three relational statements. However, an 'alloy of' some elements means there might be a predicate named alloy taking two arguments, testing if the first contains the second (which is in list form since an alloy can have more than two elements in it).

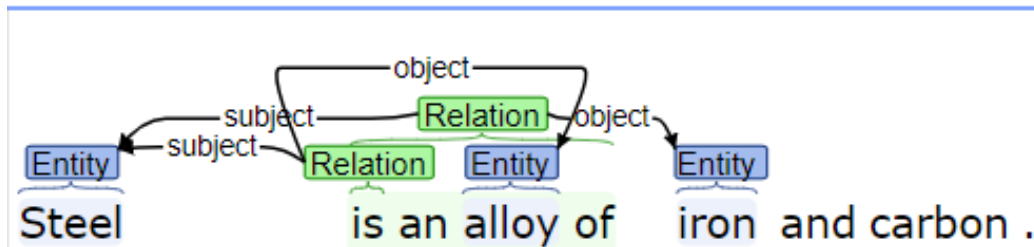


Figure 3: Predicate-argument structure for: 'Steel is an alloy of iron and carbon.'