The Semantics and Pragmatics of Natural Language

Main Concepts

1. ***Natural Language***
   - **used by human beings for communication...**
   - **sign, system, symbols, ruleset (or grammar)**
2. ***Semantics***
   - **word meaning, causes of words change ...**
3. ***Pragmatics***
   - **how language is used by a emitent in a given context, with the intention to act in a determined mode and with certain effects on the interlocutor ...**

Natural Language Processing – a subdomain of Artificial Intelligence and Linguistics

1. *Thematic Areas*

- **Linguistics - mathematical linguistics - computational linguistics**
- **Formal Language**
- **Linguistic and Language Processing**
- **The grammatical structure of utterances: the sentence, constituents, phrase, classifications and structural rules, syntactic processing ...**
- **Parser**

Formal language

1. *Symbol*

- a character, an abstract entity that has no meaning by itself

Ex: lettters, digits and special characters

2. *Alphabet*

- finite set of symbols
- often denoted by Σ

Ex: B = {0, 1} says B is an alphabet of two symbols, 0 and 1

C = {a, b, c} – C an alphabet of 3 symbols, a, b and c

Formal language

3. ***String or a word***

- **a finite sequence of symbols from an alphabet**

Ex: 01110 and 111 are strings from the alphabet B above

aaabccc and b are strings from the C above

4. ***Language***

- **a set of strings from an alphabet**

5. ***Formal language*** **(**or simply language**)**

- **a set L of strings over some finite alphabet Σ**
- **described using formal grammars**

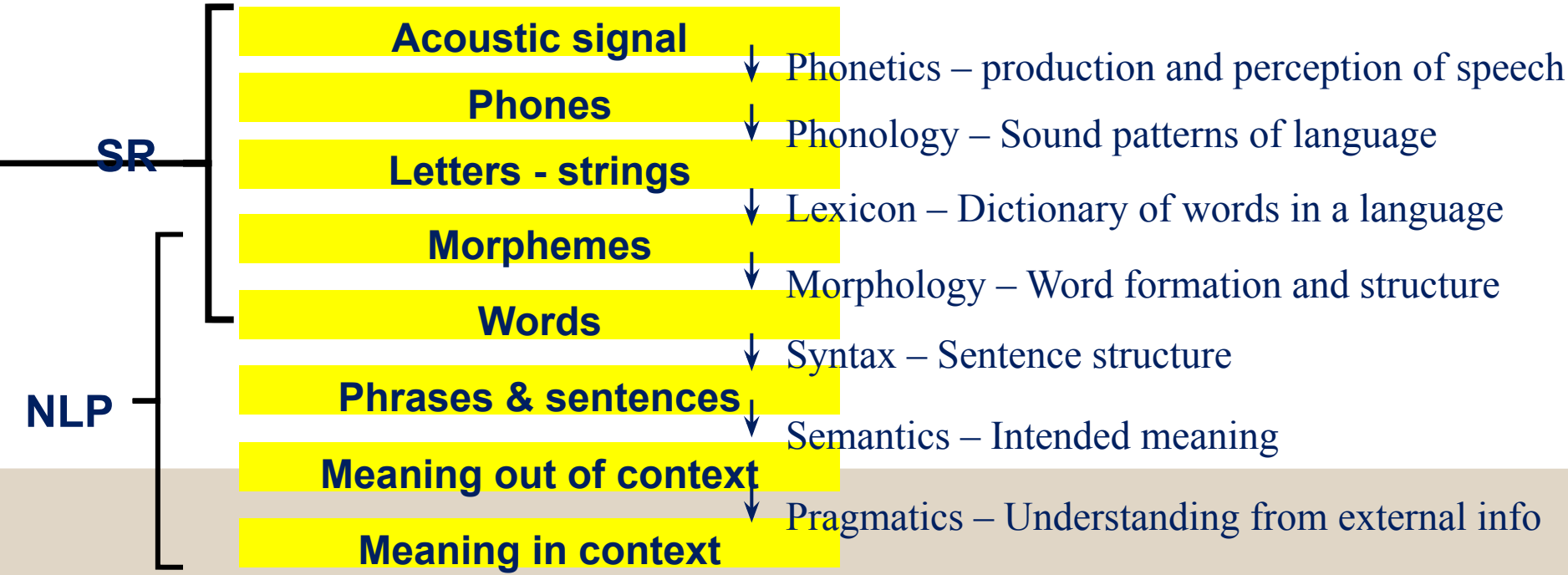
Linguistic and Language Processing

1. *Linguistics*
   - **Science of language. Includes:**

1. **Sounds (phonology)**
2. **Word formation (morphology)**
3. **Sentence structure (syntax)**
4. **Meaning (semantics) and understanding (pragmatics)…**

2. *Levels of linguistic analysis*
   - **Higher level → Speech Recognition (SR)**
   - **Lower levels → Natural Language Processing (NLP)**

Levels of Linguistic Analysis



Steps of NLP

1. ***Morphological and Lexical Analysis***

- **Lexicon**
- **Morphology – identification, analysis and description of structure of words**
- **Words – the smallest units of syntax**
- **Syntax – the rules / principles that govern the sentence structure of any language**
- **Lexical analysis – dividing text into paragraphs, sentences and words**

2. ***Syntactic analysis***

- **Analysis of words in a sentence, knowing the**

Steps of NLP

3. *Semantic Analysis*

- **Derives an absolute (dictionary definition) meaning from the context**
- **The structure created by the syntactic analyzer are assigned meaning. A mapping is made between the syntactic structure and objects in the task domain.**

Ex: “Colourless green ideas…” – correct?

4. *Discourse Integration*

- **The meaning of an individual sentence may depend on the sentences that precede it and may influence the meaning of the sentences that follow it.**

Steps of NLP

5. *Pragmatic analysis*

- **Derives knowledge from the external commonsense information**
- **Means understanding the purposeful use of language in situations particularly those aspects pf language which require world knowledge**
- **What was said is reinterpreted to determine what was actually meant.**

Ex: "Do you know what time it is" – should be interpreted as a request.

Semantics and pragmatics (S & P)

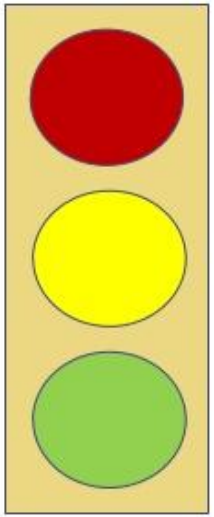
1. *S & P*

- **2 stages of analysis concerned with getting at the meaning of a sentence;**
- **1st – S – a partial representation of the meaning based on the possible syntactic structure(s) of the sentence and the meanings of the words in that sentence;**
- **2nd – P – the meaning based on the contextual and the world knowledge.**

Semantics and pragmatics (S & P)

Example: Traffic Light
Syntax – Semantics - Pragmatics

- **Syntax**
  - *green (bottom); yellow; red*
- **Semantics**
  - *green = go; …; red = stop*
- **Pragmatics**
  - If ***red*** and ***no traffic***
    then ***allowed to go***



Semantics and pragmatics (S & P)

1. ***Ex. for differences:***

"He asked for the boss".

We can work out that:

1. **Someone (who is male) asked for someone who is a boss.**
2. **We can't say who these people are and why the first guy wanted the second.**
3. **If we know something about the context (including the last few sentences spoken/written) we may be able to work these things out.**
4. **Maybe the last sentence was: "Fred had just been sacked".**
5. **From our general knowledge that bosses generally sack people: if people want to speak to people who sack them it is generally to complain about it.**
6. **We could then really start to get at the meaning of the sentence:**

Reference Resolution

A natural language expression used to perform reference is called a referring expression, and the entity that is referred to is called Referring expression.

Referent the referent. Thus, Victoria Chen and she in passage are referring expressions,

and Victoria Chen is their referent.

The discourse model contains representations of the entities that have been referred to in the discourse and the relationships in which they participate.

Thus, there are two components required by a system to successfully interpret (or produce) referring expressions:

- a method for constructing a discourse model that evolves with the dynamically-changing discourse it represents
- a method for mapping between the signals that various referring expressions encode and the hearer's set of beliefs, the latter of which includes this discourse model.

Two reference resolution tasks:
 coreference resolution
 pronominal anaphora resolution.

Coreference resolution is the task of finding referring expressions in a text that refer to the same entity,

Coreference chains that corefer. We call the set of co referring expressions a coreference chain.

Coreference resolution thus requires finding all referring expressions in a discourse and grouping them into coreference chains.

By contrast, pronominal anaphora resolution is the task of finding the antecedent for a single pronoun

Pronominal anaphora resolution can be viewed as a subtask of coreference resolution.

Different kinds of reference phenomena. We then give various algorithms for reference resolution.

Pronominal anaphora has received a lot of attention in speech and language processing, and so there are three algorithms for pronoun processing: the Hobbs algorithm, a Centering algorithm, and a log-linear (MaxEnt) algorithm.

Consider the possibilities in
example.
According to Doug, Sue just bought a 1961 Ford Falcon.
a. But that turned out to be a lie.
b. But that was false.
c. That struck me as a funny way to describe the situation.
d. That caused a financial problem for Sue.
The referent of that is a speech act in (a), a proposition in (b),
a manner of description in (c), and an event in (d). The field awaits the
development of robust methods for interpreting these types of reference.

Reference Phenomena

five types of referring expression:

- indefinite noun phrases
- definite noun phrases
- pronouns
- demonstratives
- names

Indefinite Noun Phrases

The most common form of indefinite reference is marked with the determiner *a* (or *an*), but it can also be marked by a quantifier such as *some* or even the determiner *this*:

(a) Mrs. Martin was so very kind as to send Mrs. Goddard *a beautiful goose*.

(b) He had gone round one day to bring her *some walnuts*. (c) I saw *this beautiful Ford Falcon* today.

Definite Noun Phrases

Definite reference is used to refer to an entity that is identifiable to the hearer. An entity can be identifiable to the hearer because it has been mentioned previously in the text, and thus is already represented in the discourse model:

It concerns a white stallion which I have sold to an officer. But the pedigree of *the white stallion* was not fully established.

Pronouns

Example:

Emma smiled and chatted as cheerfully as *she* could

Demonstratives

Demonstrative pronouns, like *this* and *that*, behave somewhat differently than simple definite pronouns like *it*. They can appear either alone or as determiners, for instance, *this ingredient*, *that spice*.

This and *that* differ in lexical meaning; (*this*, the **proximal demonstrative**, indicating literal or metaphorical closeness, while *that*, the **distal demonstrative** indicating literal or metaphorical distance (further away in time, as in the following example)):

I just bought a copy of Thoreau's *Walden*. I had bought one five years ago. *That one* had been very tattered; *this one* was in much better condition.

Features for Pronominal Anaphora Resolution

We now turn to the task of resolving pronominal reference. In general, this problem is formulated as follows. We are given a single pronoun (*he, him, she, her, it*, and sometimes *they/them*), together with the previous context.

Our task is to find the antecedent of the pronoun in this context. We present three systems for this task; but first we summarize useful constraints on possible referents.

We begin with five relatively hard-and-fast morphosyntactic features that can be used to filter the set of possible referents: **number**, **person**, **gender**, and **binding theory** constraints.

Number Agreement:

she/her/he/him/his/it are singular, *we/us/they/them* are plural, and *you* is un-specified for number. Some illustrations of the constraints on number agreement:

John has a Ford Falcon. It is red.

* John has a Ford Falcon. They are red.

John has three Ford Falcons. They are red.

* John has three Ford Falcons. It is red.

Person Agreement:

A first person pronoun (*I*, *me*, *my*) must have a first person antecedent (*I*, *me*, or *my*).

A second person pronoun (*you* or *your*) must have a second person antecedent (*you* or *your*).

A third person pronoun (*he, she, they, him, her, them, his, her, their*) must have a third person antecedent (one of the above or any other noun phrase).

Gender Agreement:

Referents also must agree with the gender specified by the referring expression.

English third person pronouns distinguish between *male*, (*he, him, his*), *female*, (*she, her*) and *nonpersonal* (*it*) genders.

John has a Ford. He is attractive. (he=John, not the Ford)

(John has a Ford. It is attractive. (it=the Ford, not John)

Binding Theory Constraints:

Reference relations may also be constrained by the syntactic relationships between a referential expression and a possible antecedent noun phrase when both occur in the same sentence. For instance, the pronouns in all of the following sentences are subject to the constraints indicated in brackets.

John bought himself a new Ford. [himself=John]

John bought him a new Ford. [him≠John]

John said that Bill bought him a new Ford. [him≠Bill]

John said that Bill bought himself a new Ford. [himself=Bill]

He said that he bought John a new Ford. [He≠John; he≠John]

English pronouns such as *himself*, *herself*, and *themselves* are called **reflexives**.

Selectional Restrictions:

The selectional restrictions that a verb places on its arguments may be responsible for eliminating referents, as in example:

John parked his car in the garage after driving it around for hours.

There are two possible referents for *it*, the car and the garage. The verb *drive*, however, requires that its direct object denote something that can be driven, such as a car, truck, or bus, but not a garage.

Recency:

features for predicting the referent of a pronoun that are less hard-and-fast.

Entities introduced in recent utterances tend to be more salient than those introduced from utterances further back. Thus, in example , the pronoun *it* is more likely to refer to Jim’s map than the doctor’s map.

The doctor found an old map in the captain’s chest. Jim found an even older map hidden on the shelf. It described an island.

Grammatical Role:

Many theories specify a salience hierarchy of entities that is ordered by the grammatical position of the expressions which denote them. These typically treat entities mentioned in subject position as more salient than those in object position, which are in turn more salient than those mentioned in subsequent positions.

Passages such as and lend support for such a hierarchy. Although the first sentence in each case expresses roughly the same propositional content, the preferred referent for the pronoun *he* varies with the subject in each case – John in (1) and Bill in (2)

(1) Billy Bones went to the bar with Jim Hawkins. He called for a glass of rum. [he = Billy]

(2) Jim Hawkins went to the bar with Billy Bones. He called for a glass of rum. [he = Jim]

Pronominal Anaphora Baseline: The Hobbs Algorithm

The Hobbs algorithm depends only on a syntactic parser plus a morphological gender and number checker. For this reason it is often used as a baseline when evaluating new pronominal anaphora resolution algorithms.

The input to the Hobbs algorithm is a pronoun to be resolved, together with a syntactic parse of the sentences up to and including the current sentence. The algorithm searches for an antecedent noun phrase in these trees. The intuition of the algorithm is to start with the target pronoun and walk up the parse tree to the root *S*.

For each *NP* or *S* node that it finds, it does a breadth-first left-to-right search of the node's children to the left of the target.

As each candidate noun phrase is proposed, it is checked for gender, number, and person agreement with the pronoun.

If no referent is found, the algorithm performs the same breadth-first search on preceding sentences.

The Hobbs algorithm does not capture all the constraints and preferences on pronoun - initialization described above.

It does, however, approximate the *binding theory*, *recency*, and *grammatical role* preferences by the order in which the search is performed, and the *gender*, *person*, and *number* constraints by a final check.

The steps of the **Hobbs algorithm** are as follows:

1. Begin at the noun phrase (NP) node immediately dominating the pronoun.
2. Go up the tree to the first NP or sentence (S) node encountered. Call this node X, and call the path used to reach it *p*.
3. Traverse all branches below node X to the left of path *p* in a left-to-right, breadth- first fashion. Propose as the antecedent any NP node that is encountered which has an NP or S node between it and X.
4. If node X is the highest S node in the sentence, traverse the surface parse trees of previous sentences in the text in order of recency, the most recent first; each tree is traversed in a left-to-right, breadth-first manner, and when an NP node is encountered, it is proposed as antecedent. If X is not the highest S node in the sentence, continue to step 5.
5. From node X, go up the tree to the first NP or S node encountered. Call this new node X, and call the path traversed to reach it *p*.
6. If X is an NP node and if the path *p* to X did not pass through the Nominal node that X immediately dominates, propose X as the antecedent.
7. Traverse all branches below node X to the *left* of path p in a left-to-right, breadth- first manner. Propose any NP node encountered as the antecedent.
8. If X is an S node, traverse all branches of node X to the *right* of path *p* in a left-to- right, breadth-first manner, but do not go below any NP or S node encountered. Propose any NP node encountered as the antecedent.
9. Go to Step 4.

Jack and Jill went up the hill

to fetch a pail of water.

Jack fell down and broke his crown

and Jill came tumbling after.

Now, the question is: To whom the pronoun **‘his’** refers to ?? Well to answer this, we as a human can easily relate that the word ‘his’ refers to Jack and not to the Jill, hill or the crown.

The task of locating all expressions that are coreferential with any of the entities identified in the text is known as ***coreference resolution****, and it occurs when two or more expressions in the text relate to the same person or object.*

As a result, pronouns and other referring expressions must be resolved in order to infer the correct understanding of the text.

So to perform this task computers take help of different techniques, one of which is Hobbs algorithm.

Hobbs algorithm is one of the several approaches for pronoun resolution. The algorithm is mainly based on the syntactic parse tree of the sentences. To make the idea more clear let’s consider the previous example of Jack and Jill and understand how we humans try to resolve the pronoun **‘his’.**

Jack and Jill went up the hill

to fetch a pail of water.

Jack fell down and broke his crown

and Jill came tumbling after.

As shown, the possible candidates for resolving pronoun ‘his’ were Jack, Jill, hill, water and crown.

But then why we didn’t even thought of **crown** as a possible solution?

because the noun ‘crown’ came after the pronoun ‘his’. This is the first assumption in the Hobbs algorithm, where the **search** for the referent is always **restricted to the left** of the target and hence crown is eliminated.

Then can Jill, water or hill be the possible referents?

But we know that ‘his’ may not refer to Jill because Jill is a girl.

Generally **animate objects** are referred to either by

male pronouns like- he, his; or

female pronouns like- she, her, etc. and

inanimate objects take **neutral gender** like- it..

This property is known as **gender agreement** which eliminates the possibilities of Jill, hill and water.

Pronouns can only go a few sentences back, and entities closer to the referring phrase are more important than those further away… which finally leaves us with the only possible solution i.e. Jack. This property is known as **Recency property.**

using **Hobbs algorithm**

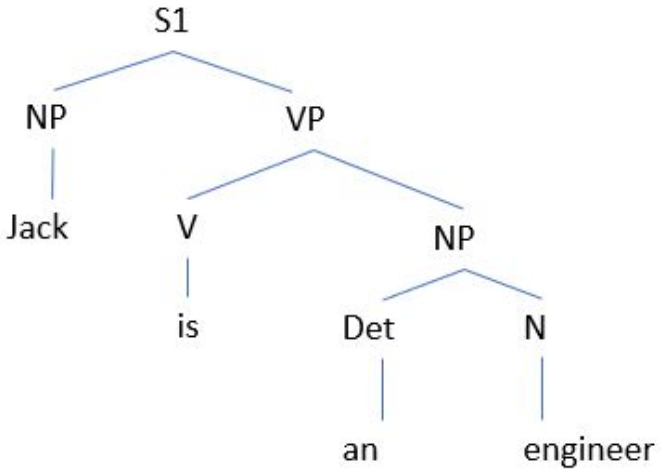
Consider two sentences:

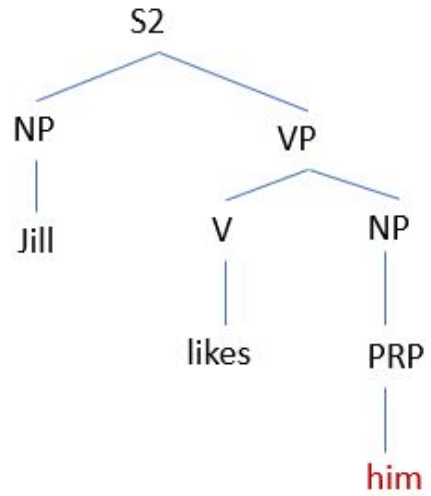
Sentence 1(S1): Jack is an engineer.

Sentence 2 (S2): Jill likes him.

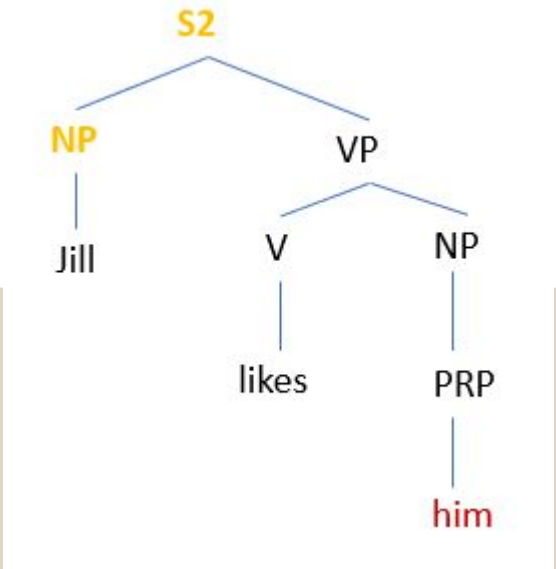
The algorithm makes use of **syntactic constraints** when resolving pronouns. The input to the Hobbs algorithm is the pronoun to be resolved together with the syntactic parse of the sentences up to and including the current sentence.

So here, we have the **syntactic parse tree** of the two sentences as shown.





The algorithm starts with the target pronoun and walks up the parse tree to the root node ‘S’. For each noun phrase or ‘S’ node that it finds, it does the **breadth first left to right search** of the node’s children to the left of the target. So in our example, the algorithm starts with the parse tree of the sentence 2 and climbs up to the root node S2. Then it does a breadth first search to find the noun phrase (NP). Here the algorithm, finds its first noun phrase for noun ‘Jill’.



But it does not explore that branch because of the syntactic constraint of **Binding theory.**

*Binding theory states that: A **reflexive** can refer to the subject of the most immediate clause in which it appears, whereas a **nonreflexive** cannot corefer this subject.. Words such as himself, herself, themselves, etc. are known as reflexive.*

Let's understand this with an example.

- John bought himself a new car.

Here, himself refers to John. Whereas if the sentence is

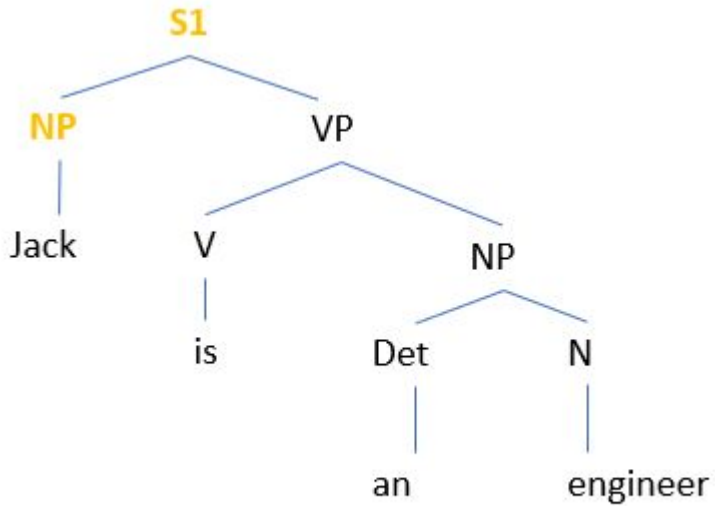
- John bought him a new car.

Then the pronoun him does not refer to John. Since one of the possible interpretation of the sentence can be John bought him a new car, where him maybe someone whom the John is gifting a car.

So according to the binding theory constraint, ‘him’ in our example will not refer to Jill.

Also because of the **gender agreement constraint** even if the branch was explored, Jill won’t be the accepted referent for pronoun ‘him’.

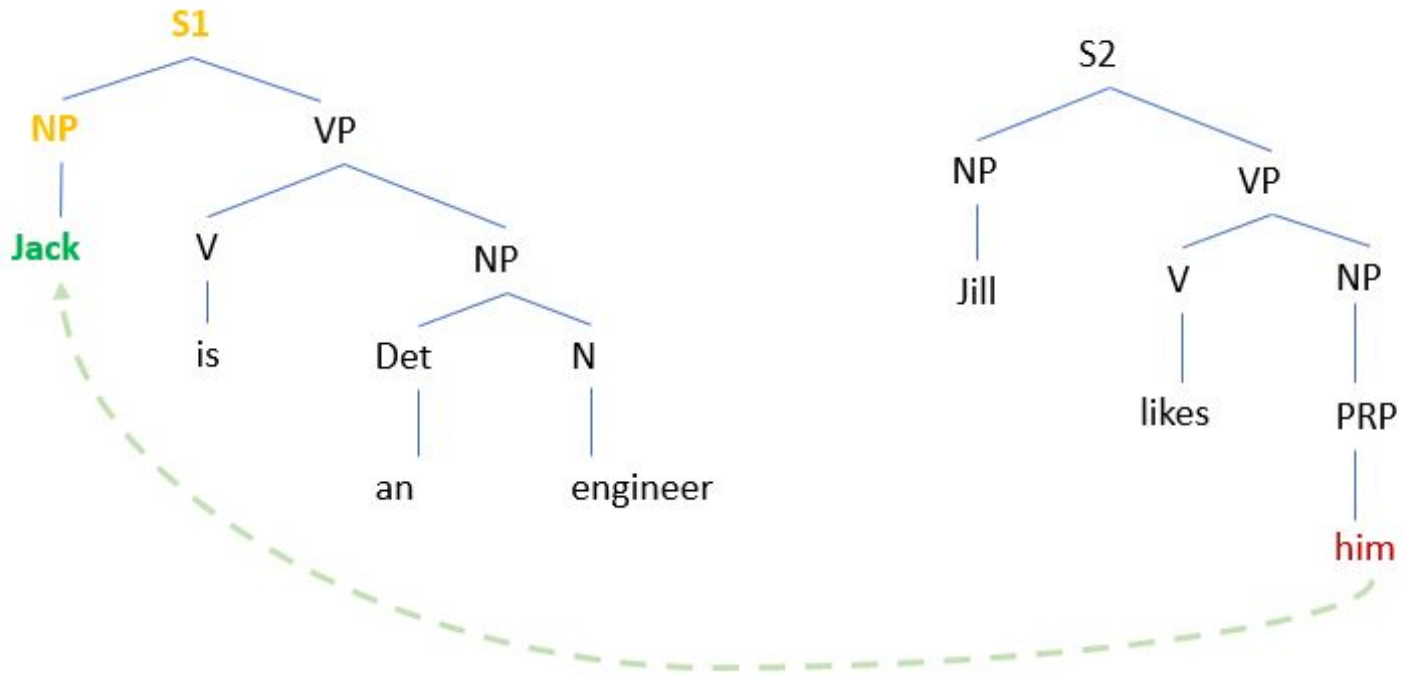
Hence the algorithm now starts the search in the syntax tree of the previous sentence.



For each noun phrase that it finds it does a breadth first **left to right** search of the node's children. This is because of the grammatical rule or more commonly known as **Hobbs distance property**.

Hobbs distance property states that entities in a subject position are more likely the possible substitute for the pronoun than in the object position.

And hence the **subject Jack** in the sentence, Jack is an engineer, is **explored before** the **object engineer** and finally Jack is the resolved referent for the pronoun him.


S1
NP
VP
Jack
V
NP
is
Det
N
an
engineer
S2
NP
VP
Jill
V
NP
likes
PRP
him