Hobbs Algorithm — Pronoun Resolution

Hobbs Algorithm is one of the technique used for **Pronoun Resolution**.

But what is Pronoun Resolution. Let's understand this with an example.

You all maybe familiar with this nursery rhyme. Read the text carefully.

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. But do you think is this task easy for computers as well ?

The answer to this is '**NO'.** Guess why ?

Because computers lack Common sense.

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 computer 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 clearer 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 think of **crown** as a possible solution? Maybe 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 **Regency property.**

Now after understanding how humans process text and resolve pronouns, let's see how we can embed intelligence (using **Hobbs algorithm**) in machines who lacks common sense, to perform the task of pronoun resolution.

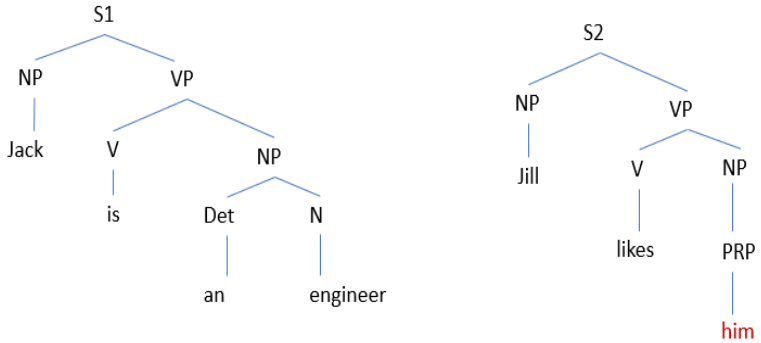
Consider two sentences:

Sentence 1(S1): Jack is an engineer.

Sentence 2 (S2): Jill likes him.

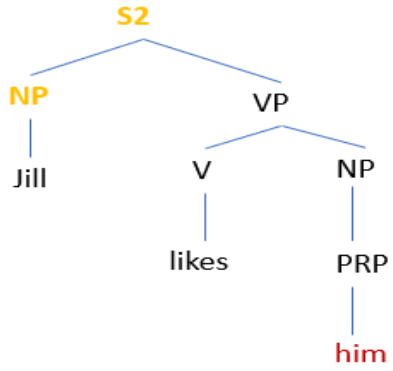
The algorithm makes use of **syntactic constraints** when resolving pronouns. The input to the Hobbs algorithm is the pronoun to be resolved together with the syntactic parse of the sentences up to and including the current sentence.

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.

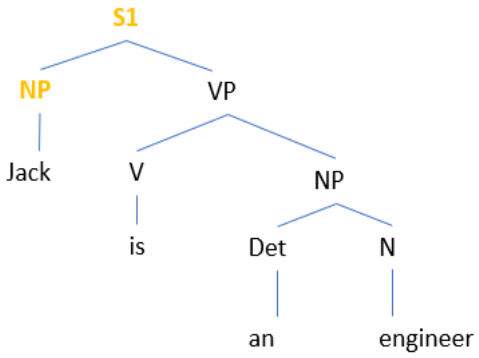
Here, himself refers to John. Whereas if the sentence is

- 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'.

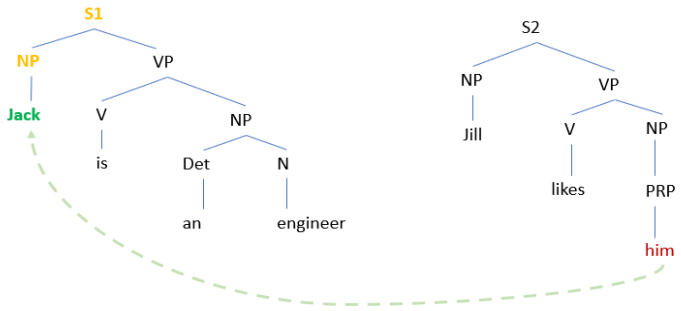
Hence the algorithm now starts the search in the syntax tree of the previous sentence.



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.



This is how the Hobbs algorithm can aid the process of pronoun resolution which is one of the crucial subtask of **natural language understanding** and **natural language generation**.