electronics

Article

Experiences on the Improvement of Logic-Based Anaphora Resolution in English Texts

Stefano Ferilli [*],[†] **and Domenico Redavid** [†]

Department of Computer Science, University of Bari, 70125 Bari, Italy; domenico.redavid1@uniba.it
[*] Correspondence: stefano.ferilli@uniba.it; Tel.: +39-080-544-2293
[†] These authors contributed equally to this work.

Abstract: Anaphora resolution is a crucial task for information extraction. Syntax-based approaches are based on the syntactic structure of sentences. Knowledge-poor approaches aim at avoiding the need for further external resources or knowledge to carry out their task. This paper proposes a knowledge-poor, syntax-based approach to anaphora resolution in English texts. Our approach improves the traditional algorithm that is considered the standard baseline for comparison in the literature. Its most relevant contributions are in its ability to handle differently different kinds of anaphoras, and to disambiguate alternate associations using gender recognition of proper nouns. The former is obtained by refining the rules in the baseline algorithm, while the latter is obtained using a machine learning approach. Experimental results on a standard benchmark dataset used in the literature show that our approach can significantly improve the performance over the standard baseline algorithm used in the literature, and compares well also to the state-of-the-art algorithm that thoroughly exploits external knowledge. It is also efficient. Thus, we propose to use our algorithm as the new baseline in the literature.

Keywords: anaphora resolution; entity resolution; information extraction

Citation: Ferilli, S.; Redavid, D. Experiences on the Improvement of Logic-Based Anaphora Resolution in English Texts. *Electronics* **2022**, *11*, 372. https://doi.org/10.3390/electronics11030372

Academic Editor: Savvas A. Chatzichristofis

Received: 15 November 2021
Accepted: 21 January 2022
Published: 26 January 2022

1. Introduction

The current wide availability and continuous increase of digital documents, especially in textual form, makes it impossible to manually process them, except for a few selected and very important ones. For the bulk of texts, automated processing is a mandatory solution, supported by research in the Natural Language Processing (NLP) branch of Artificial Intelligence (AI). Going beyond 'simple' information retrieval, typically based on some kind of lexical indexing of the texts, trying to understand (part of) a text's content and distilling it so as to provide it to end users or to make it available for further automated processing is the task of the information extraction field of research, e.g., among other objectives, it would be extremely relevant and useful to be able to automatically extract the facts and relationships expressed in the text and formalize them into a knowledge base that can subsequently be consulted for many different purposes: answering queries whose answer is explicitly reported in the knowledge base, carrying out formal reasoning that infers information not explicitly reported in the knowledge base, etc.

In fact, our initial motivation for this work was the aim of improving the performance of the tool ConNeKTion [1] in expanding automatically the content of the GraphBRAIN knowledge graph [2,3] by automatically processing the literature (e.g., texts on the history of computing [4]).

For our purposes, the system needs to know exactly who are the players involved in the facts and relationships, e.g., given the text "Stefano Ferilli works at the University of Bari. He teaches Artificial Intelligence", the extracted facts might be

```
worksAt(stefano_ferilli,university_of_bari).
teaches(he,artificial_intelligence).
```

but the latter is obviously meaningless if taken in isolation. The appropriate information we are trying to add to our knowledge base is

```
teaches(stefano_ferilli,artificial_intelligence).
```

To be able to do this, we need to understand that the generic subject 'he' is actually 'Stefano Ferilli', and replace the former by the latter before or when generating the fact.

This is a case of Entity Resolution (ER), the information extraction task aimed at addressing "the problem of extracting, matching and resolving entity mentions in structured and unstructured data" [5].

Whilst ER targets all kinds of references available in the discourse, specific kinds of references have different peculiarities and should be handled differently. Three kinds of references investigated in the literature are Anaphora, Cataphora and Coreferences. Anaphora Resolution (AR), Cataphora Resolution and Coreference Resolution are the processes of spotting those mentions and identifying the actual entity they refer to. Before delving into the AR problem specifically, which is the subject of this work, let us briefly define these terms and clarify their overlapping and differences.

Anaphora. An anaphora (from Greek 'carrying up' [6]) is "a linguistic relation between two textual entities which is determined when a textual entity (the anaphor) refers to another entity of the text which usually occurs before it (the antecedent)" [7]. Typical anaphoras are pronouns but they may take many other, less obvious, forms. Furthermore, not all pronouns are anaphoric: sometimes they are required by the language's grammatical constructs, carrying no meaning, and only the context may reveal their real nature, e.g., in sentence "John took his license when he was 18.", 'he' and 'his' are anaphoras referring to entity 'John'. Conversely, in sentence "It's raining", pronoun 'it' is not an anaphora, since it has no referred entity.

Cataphora. A cataphora (from Greek 'carrying down') is in some sense the 'opposite' of an anaphora [8]: whilst the latter references an entity located earlier in the text, the former references an entity that will be mentioned later in the discourse (typically in the same sentence). This kind of reference is more frequent in poetry, but can also be found in common language. e.g., in sentence "Were he in his twenties, John would have been eligible." , entity 'John', referenced by cataphora 'he' and 'his', is located after them in the sentence.

Coreference. Finally, according to the Stanford NLP Group, coreferences are "all expressions that refer to the same entity in a text" [9]. More specifically, Coreference Resolution is defined in [10] as "the task of resolving noun phrases to the entities that they refer to".

So, whilst anaphora and cataphora are clearly disjoint, coreferences are a proper superset of both of them [11]. ConNeKTion already includes an implementation of the anaphora resolution algorithm RAP from [12], but it uses much external knowledge about English, that may not always be useful for technical texts or available for other languages. Thus, we would like to replace it by an algorithm that is more generic and based only on the syntactic structure of the text.

As for most other NLP tasks, the specific steps and resources to carry out an ER activity strictly depend on the language in which the text is written. Different languages have very different peculiarities, some of which have a direct impact on this activity, e.g., while in English, pronouns must always be explicit, in Italian they may be elliptical, implicitly derivable from the verb thanks to the much more varied inflection of verbs than in English. This adds complexity to the task in Italian. On the other hand, in Italian it is often easier than in English to guess the gender and number of a noun or adjective, thanks to the last letter only, which is in most cases determinant to disambiguate cases in which different associations are possible structurally. These considerations were behind the aims of this work:

- Showing that a knowledge-poor, rule-based approach is viable and performant, so that it may be used to deal with languages having a more complex syntax;

- Showing that knowledge about entity gender, that may improve AR performance, may be acquired automatically also for languages where the gender is not obviously detected from morphology alone.

Carrying on a preliminary work started in [13], here we will specifically focus on AR in English, proposing an approach that extends a classical algorithm in the literature in two directions:

1. Improving the set of base rules;
2. Taking into account gender and number agreement between anaphora and referent in the case of proper nouns.

We focused on English because datasets, golden standards and baseline systems are available for it. Still, the approach should be general and easily portable to other languages. The most relevant contributions of our approach are in its ability of:

- Handling differently different kinds of anaphoras, by extending the rule set of an established baseline algorithm; and
- Disambiguating alternate associations, by using automated gender recognition on proper nouns.

The paper is organized as follows. After introducing basic linguistic information about anaphora and AR, and discussing related works aimed at solving the AR task, Section 3 describes our proposed method to improve logic-based AR. Then, Section 4 describes and discusses the experimental setting and results we obtained, before concluding the paper in Section 5.

2. Basics and Related Work

In this section, we will discuss different types of anaphora, different algorithms developed in the literature to address the AR problem, with their associated strengths and weaknesses, and suitable evaluation approaches for them. In the following, when making examples, we will adopt the convention of using italics for anaphoras and bold for the corresponding antecedents.

2.1. Anaphora and Anaphora Resolution

As said, Anaphora Resolution, aimed at finding the references corresponding to anaphoras, is a special case of Entity Resolution. In spite of their etymology, anaphora resolution is sometimes intended as encompassing cataphora resolution, e.g., Reference [14] defines it as "the problem of resolving references to earlier or later items in the discourse. These items are usually noun phrases representing objects in the real world called referents but can also be verb phrases, whole sentences or paragraphs". Additionally, Coreference Resolution is often mistaken with AR, due to their quite similar definitions and aims and to their partial overlapping. However, the difference between them [10] is apparent if we consider that two entities are co-referring to each other "if both of them resolve to a unique referent (unambiguously)" while they are anaphoric "if A is required for the interpretation of B" (differently from coreferences, it is neither reflexive nor symmetric). An example proving that AR is not a special case of CR is provided in [15]: in the sentence "**Every speaker** had to present *his* paper", 'his' is anaphoric to 'Every speaker', but it is not co-referring to it. Indeed, by replacing the pronoun with its referent, the resulting sentence "Every speaker had to present [every speaker's] paper" is semantically different from the original one: in the former, each speaker should present one paper (the one he has authored), while in the latter each speaker presents the papers of all speakers. Nor Coreference Resolution is a subset of Anaphora Resolution, due to the larger set of references solved by the former, including cataphoric references and other, even more sophisticated, ones.

In the context of AR, the candidate item to be referenced is called *anaphora*, while the item linked to the anaphora is called *reference* (or *referent*). Anaphora can be *intra-sentential*, when both the anaphora and the reference are in the same sentence, or *inter-sentential*, if the reference is located in a different sentence than the anaphora. As an example, the two sentences "**John** took *his* license when *he* was 18. *He* passed *his* exam at *his* first attempt." contain several anaphoras (various occurrences of 'he' and 'his'), all referring to entity John. The first two are intra-sentential (located in the same sentence mentioning John), the others are inter-sentential (located in a different sentence).

There are several types of anaphora, that can be classified according to the grammatical form they take. Interestingly, there are also non-anaphoric pronominal references. It is important to recognize them, so as to avoid wrongly resolving false anaphora. We will now briefly discuss each type.

2.1.1. Pronominal Anaphora

The most common type of anaphora, called pronominal anaphora, is expressed by a pronoun (as in the example about John taking his license). Pronominal anaphoras can be classified into four main groups, depending on the type of pronoun [16]: Nominative (he, she, it, they), Reflexive (himself, herself, itself, themselves), Possessive (his, her, its, their), or Objective (him, her, it, them). We would also include Relative pronouns (who, which, that, whose, whom).

Some authors consider as belonging to this category slight variations of pronominal anaphoras, including:

- *Discontinuous sets* (or 'split anaphora'), first highlighted by Mitkov in [17], where an anaphora corresponds to many entities, to be considered together as a single reference. E.g., in "**John and Mary** attended a conference. *They* say it was really inspiring.", the anaphora 'they' refers to both entities 'John' and 'Mary' as a compound entity.
- *Adjectival pronominal* anaphora [11], an anaphora that refers to an adjectival form of the entity occurred earlier in the discourse. e.g., in "**A meeting of researchers on AI** was held in Rome. *Such events* are very interesting.", the anaphora 'such events' refers to 'A meeting of researchers on AI'.

Non-Anaphoric Pronominal References

A problem affecting ER tasks is the presence of pronouns with no references associated: some of them are simply part of complex sentences and common expressions. Three kinds of references fall into these kinds of non-anaphoric pronouns:

- **Extrapositions** are transformations of the texts that affect clauses to be moved (extraposed) earlier or later in the discourse, referencing the subject of the clause with the pronoun 'it'. "It is well-known that in Winter colds are more frequent." 'It' has no corresponding entity as a reference, it just refers to the fact about colds as a whole.
- **Clefts** are sentences expressing their meaning using a construction more complex than needed, in order emphasize something. "It was John who wrote the code." 'It' has no reference, it's just there to emphasize John in the sentence meaning, that is simply "John wrote the code", giving him the prize or guilt of doing that.
- **Pleonastic 'it'** is a dummy pronoun with no actual reference, commonly used in natural language, used in sentences in which there is no subject carrying out the action, e.g., in "It's raining.", 'it' is only needed to express a statement on the weather conditions but there is no one who "is raining".

2.1.2. Noun Phrases

A less frequent type of anaphora is expressed by noun phrases. The most common cases in this group are Definite Noun Phrase Anaphora, where the anaphora to be referenced is a noun phrase preceded by the definite article 'the'. "**Climate change** is endangering the Earth. *The problem* is well-known". This is one of the most difficult kinds of anaphora to spot since lots of definite noun phrases in the text can have this role, and identifying them requires understanding the semantics of the text (in this case, the knowledge that climate change is a problem).

An even more subtle variant of noun phrases is the Zero Anaphora [11], in which the anaphora is not necessarily definite. A hint to spot this kind of anaphora is knowing that there is always a gap (a punctuation mark like colon, semicolon, period, parenthesis, etc.) between the anaphora and the reference. "You have two **advantages**: *your skill* and *your experience*." Both anaphoras refer to the same antecedent.

A more ambiguous kind of noun anaphora than the previous ones is the Bridging Anaphora [11]. It is based on a semantic relation existing between a noun phrase and an antecedent entity. There is no way to spot this relation without knowing about that relation, e.g., in "I wanted to run **that program**, but I knew that *the sort procedure* was incomplete", the anaphora refers to a component of the reference, and the bridge is represented by the implicit fact that the specific program includes a sort procedure (including procedures is a semantic property of programs).

2.1.3. Other Anaphoric References

There are cases, not considered by the previous groups, that are still anaphoric when not accompanied by other terms in the same noun phrase. These anaphoras are primarily differentiated according to their lexical role, among which:

- Indefinite pronouns (presuppositions) (all, no, one, some, any, more, most, a lot, lots, enough, less, few, a little, etc.). "**The computer** is not powerful enough. He should buy a new *one*."
- Ordinals (first, second, third, last, etc.). "John won't spend any more money in **computers**. This is *the third* he buys in a year."
- Demonstratives (this, that, these, those). "If John could run his program on this **computer**, he should be able to run it on *that* too."

2.2. Anaphora Resolution Algorithms

We will refer to a recent survey on anaphora resolution algorithms reported in [11], selecting some of the approaches to be discussed in more detail based on their closer relationship to the solution we are going to propose in this paper.

Traditionally, approaches to anaphora resolution are rule-based. Whilst more recently approaches based on Neural Networks and Deep Learning (CNN, LSTM, 2D-CNN, transformers) have been proposed, this paper is specifically aimed at showing the behavior, performance and strengths of rule-based approaches to AR. In fact, they carry several advantages:

- Not depending on the distance between the anaphora and its reference, since general rules working on the syntactic structure text are used;
- Not requiring huge annotated training sets, just a linguistic expert to formalize the rules for AR in the language;
- Being explainable, since they work in the same way as humans do (also allowing application for educational purposes).

For this reason, we will not delve further into sub-symbolic or deep approaches to AR.

Rule-based algorithms can be associated with three major philosophies that have inspired improvements and variants:

Syntax-based approach which "works by traversing the surface parse trees of the sentences of the texts in a particular order". The representative of this philosophy is Hobbs' algorithm [18].

Discourse-based approach which relies on spotting "the focus of the attention, choice of referring expression and perceived coherence of utterances within a discourse segment". The representative of this philosophy is Grosz et al.'s Centering Theory [19].

Hybrid approach which relies on combining the previous two approaches. The representative of this philosophy is Lappin and Leass' algorithm [12], that determines the best reference according to its salience value, which depends on recency and grammatical function.

Another relevant distinction is between knowledge-rich and knowledge-poor approaches. Whilst the former rely on external knowledge and resources, the latter try to exploit, as far as possible, only the text itself and the syntactic information that can be derived from it. Most of the algorithms are knowledge-rich, or have optional knowledge-rich components, to improve their performance. This poses the additional problem of obtaining these resources, that are not always available or of sufficient quality for all languages [20]. One of the latest systems based on the knowledge-rich approach is COCKTAIL [21], using WordNet [22] as the external resource. WordNet is a manually developed lexical ontology for English; similar initiatives exist for a few other prominent languages, but do not always match the quality of the original. The most prominent representative of knowledge-poor algorithms is CogNIAC [23]. It adopts a minimalistic approach that focuses on specific types of references, trading recall for higher precision (see Section 2.3 for an introduction to these metrics).

Liang and Wu [16] have learnt the lesson from all the previous works, developing a system based on heuristic rules and on checking several properties based on the ideas of both Lappin and Leass (sentence recency) and the Centering Theory. They improved performance using number, gender and animacy agreement, exploiting WordNet as COCKTAIL previously did, and carrying out pleonastic 'it' detection. After Liang and Wu's work, the research interest started to shift towards Coreference Resolution since it addresses a broader, but different scope than the AR task.

Hobbs' algorithm [18] is widely recognized as one of the most powerful baseline approaches in the rule-based strategies. It has been considered as a reference for comparison by every new approach or improvement of older approaches because of its performance. For this reason, we will take it as the baseline to evaluate performance of our approach, as well. While our approach is much simpler, we will compare it also to Liang and Wu's approach, to check how much we can approach that state-of-the-art performance without bringing to bear so much power or requiring so much previous knowledge. We will now describe in some more detail these approaches.

2.2.1. Hobbs' Naïve Algorithm

Hobbs' algorithm, purposely intended for the AR task, was proposed in 1978 [18]. It works on the parse trees of the sentences in the text, expressing the syntactic structure of the word sequences that make them up in terms of Part-of-Speech (PoS) tags, using as a standard reference the Penn Treebank tagset [24]. We can distinguish two kinds of nodes in the hierarchical structure of the parse tree:

- Internal nodes, that contain PoS tags representing the sub-tree rooted in them;
- Leaf nodes, associated with simple tokens (elementary textual parts of the sentence).

Figure 1 shows the parse tree of the two sentences about John taking his license, proposed in Section 2.1.

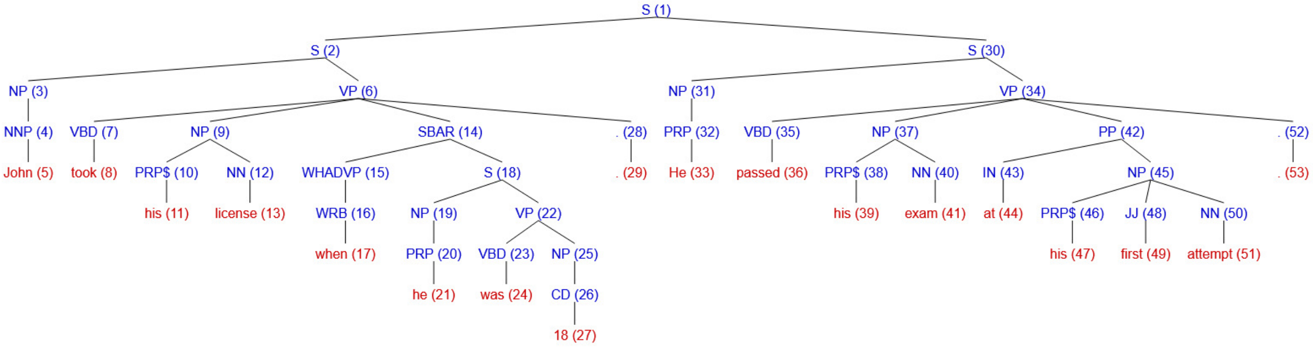

Figure 1. Sample parse tree for two sentences with several anaphora.

The most relevant PoS tags for our purposes are 'S', for sentences, and 'NP', for noun phrases. For each anaphora, the algorithm traverses the tree looking for the reference under the NP branches, following these steps:

1. Starting from the NP node immediately dominating the pronoun, climb the tree until the first NP or S. Call this node X, and call the path p.
2. Traverse left-to-right, breadth-first, all branches under X to the left of p. Propose as reference any NP that has an NP or S between it and X.
3. If X is the highest S in the sentence, traverse the parse trees of previous sentences in order of recency. Traverse each tree left-to-right, breadth-first. When encountering an NP, propose it as a candidate reference. If X is not the highest node, go to step 4.
4. From X, climb the tree until the first NP or S. Call it X, and p the path to X.
5. If X is an NP and p does not pass through the node N' immediately dominated by X, propose X as a candidate reference.
6. Traverse left-to-right, breadth-first, all branches below X to the left of p . Propose any NP encountered as a candidate reference.
7. If X is an S node, traverse all the branches of X to the right of p but do not enter the subtree under any NP or S encountered. Propose any NP as a candidate reference.
8. Go to step 3.

The algorithm involves two main sections:

- The former (steps 1–2) climbs the syntactic tree for the first time and explores it, to find inter-sentential candidates in past sentences;
- the latter (steps 4–7) continues climbing, seeking for a new NP, checking if a potential NP can be the antecedent or just exploring a new path.

These two sections are delimited by steps (3) and (8), that iterate the process and eventually seek for the first NP encountered in the analysis of the past sentence nearest to the one just visited. Let us show the practical application of the algorithm for the five pronouns in the text in Figure 1.

- Pronoun 'his' (11)
    1. The NP node immediatly dominating the pronoun is (9). Climbing up, the first NP or S is node S (2), with $p = \langle(2),(6),(9),(10),(11)\rangle$
    2. The left-to-right, breadth-first, traversal of all branches under S (2) to the left of p involves nodes $\langle(3),(4)\rangle$ with only one candidate, NP (3).
    3. (2) is highest S node in sentence, but there is no previous sentence.
    4. N/A
    5. N/A
    6. N/A
    7. No branches of X to the right of p.
    8. N/A
- Pronoun 'he' (21)
    1. The NP node immediately dominating the pronoun is (19). Climbing up, the first NP or S is node S (18) with $p = \langle(18),(19),(20),(21)\rangle$
    2. No branch to the left of S (18) to traverse.
    3. S (18) is not the highest S node in sentence,
    4. Go up to S (2) node, now $p = \langle(2),(6),(14),(18),(19),(20),(21)\rangle$.
    5. N/A
    6. The left-to-right, breadth-first, traversal of all branches under S (2) to the left of p involves nodes $\langle(3),(4)\rangle$ with only one candidate, NP (3).
    7. No branches of X to the right of p.
    8. N/A
- Pronoun 'He' (33)
    1. The NP node immediately dominating the pronoun is (31). Climbing up, the first NP or S is node S (30) with $p = \langle(30),(31),(32),(33)\rangle$.

2. No branch to the left of S (30) to traverse.
3. S (30) is not the highest S node in sentence, go to step 4.
4. Climbing from X = S (30), the first S or NP node is S (1),
   now $p = \langle(1),(30),(31),(32),(33)\rangle$.
5. N/A
6. The left-to-right, breadth-first, traversal of all branches under S (1) to the left of p, involves nodes $\langle(2),(3),(6),(4),(7),(9),(14),(28),(10),(12),(15),(18),(16),(19),(22),(20),(23),(25),(26)\rangle$, with four candidates: NP (3), NP (9), NP (19), NP (25).
7. No branches of X to the right of p.
8. No further previous sentences, stop.

- Pronoun 'his' (39)
  1. The NP node immediately dominating the pronoun is (37). Climbing up, the first NP or S is node S (30) with $p = \langle(30),(34),(37),(38),(39)\rangle$.
  2. The left-to-right, breadth-first, traversal of all branches under S (30) to the left of p, involves nodes $\langle(31),(32)\rangle$, with only one candidate, NP (31).
  3. S (30) is not the highest S node in sentence, go to step 4.
  4. Climbing from X = S (30), the first S or NP node is S (1),
     now $p = \langle(1),(30),(34),(37),(38),(39)\rangle$.
  5. N/A
  6. The left-to-right, breadth-first, traversal of all branches under S (1) to the left of p, involves nodes $\langle(2),(3),(6),(4),(7),(9),(14),(28),(10),(12),(15),(18),(16),(19),(22),(20),(23),(25),(26)\rangle$, with four candidates: NP (3), NP (9), NP (19), NP (25).
  7. No branches of X to the right of p.
  8. No further previous sentences, stop.
- Pronoun 'his' (47)
  1. The NP node immediately dominating the pronoun is (45). Climbing up, the first NP or S is node S (30) with $p = \langle(30),(34),(42),(45),(46),(47)\rangle$.
  2. The left-to-right, breadth-first, traversal of all branches under S (30) to the left of p, involves nodes $\langle(31),(32)\rangle$, with only one candidate, NP (31).
  3. S (30) is not the highest S node in sentence, go to step 4.
  4. Climbing from X = S (30), the first S or NP node is S (1),
     now $p = \langle(1),(30),(34),(42),(45),(46),(47)\rangle$.
  5. N/A
  6. The left-to-right, breadth-first, traversal of all branches under S (1) to the left of p, involves nodes $\langle(2),(3),(6),(4),(7),(9),(14),(28),(10),(12),(15),(18),(16),(19),(22),(20),(23),(25),(26)\rangle$, with four candidates: NP (3), NP (9), NP (19), NP (25).
  7. No branches of X to the right of p.
  8. No further previous sentences, stop.

It is apparent that this algorithm is only based on the grammatical role of the sentence components, completely ignoring gender and number agreement of anaphora and referent. This information would be of great help to disambiguate some candidate references, but would require external information and thus would transform the approach into a knowledge-rich one. This issue was considered by Hobbs himself in [18] as a direction for improving his algorithm, and has been an important source of performance improvement for all subsequent works on anaphora resolution (e.g., [12,16,19,21,23,25]).

Indeed, Hobbs subsequently proposed a semantic algorithm that relies on various selectional constraints based on the impossibility of action (e.g., dates cannot move, places cannot move, large fixed objects cannot move, etc.). For our work we started from the basic (naïve) version, rather than the improved one, because we want our results to be primarily derived by rules applied to the grammatical structure of the sentence, so as to be as general as possible. This choice is motivated by the fact that relying on lots of specific semantic rules might cause overfitting, since these kinds of rules are highly specific to their very small range of action. Moreover, the two versions were compared by Hobbs himself, showing that the performance achieved by the semantic algorithm for the adopted metric (Hobbs' metric) is just +3.4% on the manually evaluated texts, which is reasonably low considering that the assessed performance of the base algorithm already reached 88.3%.

2.2.2. Liang and Wu's Approach

Liang and Wu's system [16] was proposed in 2004 as an automatic pronominal anaphora resolution system for English texts. Initially aimed at accomplishing its task using heuristic rules, exploiting the WordNet ontology and obtaining further information about gender and number, its accuracy was subsequently improved by extracting information about animacy and handling pleonastic 'it' pronouns.

The process carried out by this system consists of a pipeline of steps. Once the raw text is acquired, it undergoes PoS tagging and an internal representation is built. An NP finder module finds all the candidate anaphoras to be solved. All pleonastic 'it' pronouns are excluded from the processing by a specific module. Each remaining anaphora generates a candidate set of references to which number agreement is applied. After that, they undergo the gender agreement and animacy agreement checks, leveraging the support provided by WordNet. The agreeing candidates are evaluated by heuristic rules, classified into preference and constraint rules, and the final decision is entrusted to a scoring equation dependent on the rule premises, consequences and the amount of agreement to that rule. The employed heuristic rules include syntactic and semantic parallelism patterns, definiteness ('the' + NP to address people—e.g., "the good programmer" is not an indefinite noun), mention frequency, sentence recency (Lappin and Leass' algorithm highly regarded this factor), non-propositional noun phrase rule (in Lappin and Leass' algorithm, the ranking is: subject > direct object > indirect object) and conjunction constraint (conjunct noun phrases cannot refer to each other).

2.3. Evaluation

ER/AR can be considered as a kind of information retrieval task, where the queries are the anaphoras and the results are the references. So, straightforward performance evaluation metrics would be Precision:

$$P = \frac{TP}{TP + FP}$$

and Recall:

$$R = \frac{TP}{TP + FN}$$

expressing, respectively, the ratio of correct answers among the answers given and the ratio of correct answers over the real set of correct answers, in terms of parameters TP (True Positives, the number of items correctly retrieved), FP (False Positives, the number of items wrongly retrieved), FN (False Negatives, the number of items wrongly discarded) and TN (True Negatives, the number of items correctly discarded).

These metrics require all correct answers for the dataset (the ground truth or golden standard) to be known, and they ignore the fact that, in AR, the queries themselves are not known in advance but the system itself is in charge of identifying the anaphoras (and thus it may misrecognize both the candidate anaphoras, in the first place, and their associated reference, subsequently). For this reason, Hobbs also introduced in [18] a measure which gives insights regarding the precision of AR algorithms specifically:

$$H = \frac{\text{\#correct resolutions}}{\text{\#attempted resolutions}}$$

In fact, this metric has been widely adopted in the literature, including the work by Liang and Wu which we will use for comparison.

A section in survey [11] is purposely devoted to the available dataset in the literature for generic reference resolution. It also includes a discussion on, and a comparison of, the datasets employed by the other research works in the field. The most important publicly available datasets for AR are:

- The Automatic Content Extraction (ACE) corpus [26], developed between 2000 and 2008, containing news-wire articles and labelled for different languages, including English;
- The Anaphora Resolution and Underspecification (ARRAU) corpus [27], developed around 2008 as a combination of several corpora, namely TRAINS [28,29], English Pear [30], RST [31] and GNOME [32].

Both are available through the Linguistic Data Consortium (https://www.ldc.upenn.edu/, accessed on 10 November 2021), but are not free. On the other hand, almost all previous works in the field of AR use books, magazines, manuals, narratives, without specific references. Exceptions are Hobbs [18] and Liang and Wu [16], that both use the Brown Corpus [33] (http://icame.uib.no/brown/bcm.html, accessed on 10 November 2021) for evaluating their algorithms (differently from Liang and Wu, Hobbs uses many sources, including part of this dataset). The main issue with this corpus is that it is not an AR corpus strictly speaking, i.e., with annotated anaphoras, but just an American English corpus, which means that it needed a preliminary annotation step for AR purposes.

The Brown University Standard Corpus of Present-Day American English (or simply Brown Corpus) is a linguistic dataset initially compiled by Kučera and Francis in the 1960s and updated several times until 1979. It consists of 500 samples with 2000+ words each, for a total of 1,014,312 words. Samples are divided into 15 different genres, identified by codes and belonging to two main categories:

- Informative prose (374 samples), including:
  – Press texts (reportage, editorial, review);
  – Books and periodicals (religion, skills and hobbies, popular lore, belles lettres, biography, memoirs);
  – Government documents and other minorities (foundation and industry reports, college catalog, industry house organ);
  – learned texts (natural sciences, medicine, mathematics, social and behavioral sciences, political science, law, education, humanities, technology and engineering).
- Imaginative prose (126 samples), including:
  – Novel and short stories (general fiction, mystery and detective fiction, science fiction, adventure and western fiction, romance and love story, humor).

Table 1 reports the performance of the most relevant AR systems as reported in [11], with notes on the experimental setting used, including the dataset and metrics. For the features, we use abbreviations 'Sn' for Syntax, 'D' for Discourse, 'M' for Morphology, 'Sm' for Semantics, and 'Sr' for Selectional rules. Both Hobbs and Liang and Wu used the Brown Corpus as the experimental dataset, and evaluated performance using the Hobbs' metric. Only the basic approach by Hobbs uses just syntax.

Table 1. Anaphora Resolution algorithms performance as reported in [11].

Approach	Dataset	Performance	Features
Hobbs [18]	Brown (part), Books, Magazines, fiction and non-fiction	$H = 88.3\%$ $H = 91.7\%$	Sn Sn, Sr

Lappin and Leass [12]	Computer Science manuals	H = 74% (inter-sent.)
		H = 89% (intra-sent.)
			Sn, D, M, Sm
			Sn, D, M, Sm

Centering Theory [25]	fiction and non-fiction books from [18], others	H = 77.6%	D
CogNIAC [23]	Narratives MUC-6	$P = 92\%, R = 64\%$ $P = 73\%, R = 75\%$	D, Sn D, Sn
Liang and Wu [16]	Brown (random)	H = 77%	Sm, D, Sn

3. Proposed Algorithm

The AR strategy we propose extends the original algorithm by Hobbs. When deciding the starting point in developing our strategy, we could choose any of the two main AR approaches known in the literature, i.e., the syntax-based approach (Hobbs) or the discourse-based approach (Centering Theory). We opted for the former because it emulates the conceptual mechanism used by humans to find the correct references for anaphoric pronouns, expressed in the form of grammatical rules. Furthermore, Hobbs' algorithm is still highly regarded in AR literature as a strong baseline for comparisons, given its simplicity, ease of implementation and perfomance too [11]. On the other hand, we left mixed (syntactic and discourse-based) approaches, like Lappin and Leass', for possible future extension of the current algorithm. In fact, Lappin and Leass' ideas have already been exploited for many works, while attempts at improving Hobbs' algorithm has been often neglected, which further motivated our choice.

In a nutshell, we propose a syntax-based algorithm for AR that takes Hobbs' naïve algorithm as a baseline and extends it in 2 ways:

1. Management of gender agreement on proper nouns. Gender can be associated to adjectives and nouns, and in the latter case to common or proper nouns, while common nouns can be found in dictionaries and thesauri, there are less obvious standard resources to obtain the gender of proper nouns. We propose the use of rules and pattern matching, using models built by Machine Learning algorithms, starting from a training set of first names whose gender is known and using the last letters of such names (i.e., their suffixes of fixed size) as the learning features.
2. Refinement of Hobbs rules. Hobbs' algorithm adopts a "one size fits all" perspective, trying to address all anaphora typologies with a single algorithm, but it fails on possessive and reflexive pronouns when looking for their reference intra-sententially: the subject side is never accessed. This flaw has been successfully corrected in our rules.

We consider our study on proper noun gender recognition to be our main novel contribution to the landscape of AR. Our refinement of the rules in Hobbs' algorithm should also be relevant.

3.1. GEARS

We called our approach GEARS, an acronym for 'Gender-Enhanced Anaphora Resolution System'. It takes as input a (set of) plain text(s), and returns a modified version of the original text(s) in which the anaphoras have been replaced by their referents. Algorithm 1 describes the overall processing workflow carried out by GEARS.

Algorithm 1 GEARS.

```
Require: set of documents $C$; window size $w$
  for all documents (sequences of sentences) to be processed $d = \langle s_1, \ldots, s_n \rangle \in C$ do
    for all $i = 1, \ldots, n$ (sentences in $d$) do
      resolve all anaphoras in $s_i$ ($i$-th sentence in $d$) using as the sliding window of
      sentences $\langle \mathrm{fol}(\mathrm{parse}(s_{i-w+1})), \mathrm{fol}(\mathrm{parse}(s_{i-w+2)}), \ldots, \mathrm{fol}(\mathrm{parse}(s_i)) \rangle$
    end for
  end for
```

Each document is processed separately, since an anaphora in a document clearly cannot refer an entity in another document. Furthermore, to delimit the search space for references, and ensure scalability, each document is actually processed piecewise, each piece consisting of a sliding window of a few sentences. GEARS is applied iteratively to each sentence in the text, providing a fixed-size window of previous sentences, whose size is a parameter to our algorithm. At each iteration the window is updated, by adding the new sentence to be processed and removing the oldest one (the first) in the current window.

Actually, GEARS does not work on the plain text of the sentences, but on a logical representation (obtained by applying function ’fol’ in the algorithm) of their parse tree (obtained by applying function ’parse’ in the algorithm).

So, when moving the sliding window, the new sentence to be added undergoes PoS tagging and its parse tree is extracted. As said, terminal (leaf) nodes in these trees are literals that represent a textual part of the sentence, while non-terminal nodes (internal ones and the root) are PoS tags indicating a phrase type or a syntactic part of the discourse. Then, the parse tree is translated into a First-Order Logic formalism to be used by the rule-based AR algorithm. Each node in the parse tree is assigned a unique identifier and the tree is described as a set of facts builts on 2 predicates: `node/2`, reporting for each unique node id the corresponding node content (PoS tag or literal), and `depends/2`, expressing the syntactic dependencies between pairs of nodes in the parse tree (i.e., the branches of the tree). Figure 2 reports an example of FOL formalization for the sentences concerning John and his license, whose parse tree was shown in Figure 1. During processing, facts built on another predicate, `referent/2`, are added to save the associations found between already resolved anaphoras (first argument) and their corresponding referents (second argument). We assign named tags to punctuation symbols found in the tree nodes as well, since they are not associated with names from the core parser.

```
node(1,s).
node(2,s).              depends(2,1).      node(30,s).            depends(30,1).
FL node(3,np).          depends(3,2).      node(31,np).           depends(31,30).
node(4,nnp).            depends(4,3).      node(32,prp).          depends(32,31).
node(5,’John’).         depends(5,4).      node(33,’He’).         depends(33,32).
node(6,vp).             depends(6,2).      node(34,vp).           depends(34,30).
node(7,vbd).            depends(7,6).      node(35,vbd).          depends(35,34).
node(8,’took’).         depends(8,7).      node(36,’passed’).     depends(36,35).
node(9,np).             depends(9,6).      node(37,np).           depends(37,34).
node(10,’prp$’).        depends(10,9).     node(38,’prp$’).       depends(38,37).
node(11,’his’).         depends(11,10).    node(39,’his’).        depends(39.38).
node(12,nn).            depends(12,9).     node(40,nn).           depends(40,37).
node(13,’license’).     depends(13,12).    node(41,’exam’).       depends(41,40).
node(14,sbar).          depends(14,6).     node(42,pp).           depends(42,34).
node(15,whadvp).        depends(15,14).    node(43,in).           depends(43,42).
node(16,wrb).           depends(16,15).    node(44,’at’).         depends(44,43).
node(17,’when’).        depends(17,16).    node(45,np).           depends(45,42).
node(18,s).             depends(18,14).    node(46,’prp$’).       depends(46,45).
node(19,np).            depends(19,18).    node(47,’his’).        depends(47,46).
node(20,prp).           depends(20,19).    node(48,jj).           depends(48,45).
node(21,’he’).          depends(21,20).    node(49,’first’).      depends(49,48).
node(22,vp).            depends(22,18).    node(50,nn).           depends(50,45).
node(23,vbd).           depends(23,22).    node(51,’attempt’).    depends(51,50).
node(24,’was’).         depends(24,23).    node(52,full_stop).    depends(52,30).
node(25,np).            depends(25,22).    node(53,’.’).          depends(53,52).
node(26,cd).            depends(26,25).
node(27,18).            depends(27,26).
node(28,full_stop).     depends(28,2).
node(29,’.’).           depends(29,28).
```

Figure 2. First-Order Logic facts expressing the parse tree of sample sentences.

To date, as pre-processing is concerned. Then, actual processing takes place along two main phases:

1. **Anaphora Detection** from the acquired parse trees of the sentences, pronouns are extracted and compared to the already resolved pronouns.
2. **Anaphora Resolution** for anaphoric pronouns discovered in the previous phase, the number is assessed, and the gender is assessed only for singular pronouns, that might be associated with proper nouns. Then, the rules for AR are applied to all of

them in order to find their referents, ensuring that number (and gender for singular anaphoras) match. Note that the referent of an anaphora can in turn be an anaphora, generating a chain of references. In such a case, since the previous references must have been resolved in previous iterations, the original (real) referent is recursively identified and propagated to all anaphoras in the chain.

For each anaphora detected in phase 1, the activities of phase 2 are carried out by a Pattern-Directed Inference System specifying the rule-based AR algorithm described in detail in Sections 3.1.1 and 3.1.2. It works on the facts in the logical representation of the sentences in the sliding window and, for each identified anaphora, it returns a set of 4-tuples of the form:

$$Q = \langle SA, A, R, SR \rangle$$

where SA represents the sentence in which the anaphora is found, A represents the anaphora itself, R represents the referent (or '–' if no referent can be found), and SR represents the sentence in which the referent is found (or '–' if no referent can be found). The 4-tuples corresponding to resolved anaphoras (i.e., anaphoras for which a referent is identified) are added, along with additional operational information, to a so-called 'GEARS table' associated with the text document under processing. Since each anaphora can have only a single referent, the pair $\langle SA, A \rangle$ is a key for the table entries.

Finally, when the generation of the GEARS table is complete, a post-processing phase is in charge of using it to locate in the text the actual sentences including the resolved anaphoras and replacing the anaphoras by their referents found in the previous step, so as to obtain the explicit text. Since the AR core cannot 'see' the actual sentences as plain texts (it only sees their parse trees), it must regenerate the text of SA and SR by concatenating the corresponding leaf nodes in their parse trees. The sentence in the 4-tuple acts as a pattern that, using regular expressions, is mapped onto the parts of text that correspond to the words in the leaves of the parse trees. Then, find and replace methods are applied to the original text, based again on the use of regular expressions. The updated texts are saved in a new file.

A graphical representation of the overall workflow is shown in Figure 3. From the set of documents on the far left, one is selected for processing and the sliding window (red squares) scans it, extracting the parse tree of sentences and identifying pronouns (red circles in the document and parse trees). Then, our AR strategy (1) processes all these anaphora, distingushing them into singular or plural, and then further distinguishing singular ones into proper nouns and others. Pronouns that cannot be associated with any referent are considered as non-anaphoric. Among singular anaphoras, those referring to proper nouns are identified and disambiguated relying on the model for recognizing the gender of proper nouns (2) automatically obtained using Machine Learning approaches. We will now provide the details of our components (1) and (2).

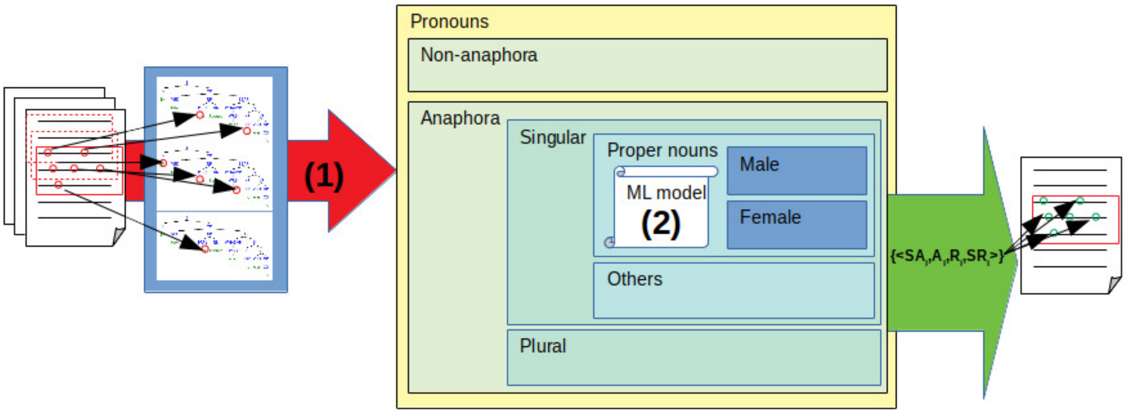

Figure 3. A scheme of the GEARS workflow, with highlighted the 2 contributions proposed in this paper: improved rules for anaphora resolution (1) and proper noun gender recognition (2).

3.1.1. Gender and Number Agreement

GEARS checks gender and number agreement using a set of ad hoc rules, applied separately to the anaphoras and the referents. Priority is given to the number of anaphoras, firstly because the anaphora is found and analyzed before the referent. If its number is plural, then we do not check the gender attribute, since plural nouns of different genders together are infrequent and, in any case, our algorithm will give priority to the closer one. So, the chances of failure in this situation are low.

Assessment of Number of Anaphoras

Each anaphora is located in a sub-tree rooted in an NP. If such NP node has a pronoun child, then its number is easily assigned based on the pronoun's number:

- 'Singular' for pronouns he, him, his, himself, she, her, hers, herself, it, its, itself;
- 'Plural' for pronouns they, them, their, theirs, themselves.

Assessment of Number of Referents

Like anaphoras, each referent is located in a sub-tree roted in an NP. The number is assigned to the NP node primarily based on its child in the tree, using the following rules (listed by decreasing priority).

- If the child of the NP node is:
  - A plural generic noun (e.g., 'computers'), or
  - A plural proper noun (e.g., 'the United States'), or
  - A singular noun with at least one coordinative conjunction (e.g., 'the computer and the mouse', 'John and Mary'),

  then the number for referent is plural;
- If the child of the NP node is:
  - A singular generic noun (e.g., 'computer'), or
  - A singular proper noun (e.g., 'John'), then the number for referent is singular;
- If the child of the NP node is a pronoun, then the corresponding number is determined using the rules for anaphoras described in the previous paragraph.

Assessment of Gender of Singular Anaphoras

The gender of (singular) anaphoras is easily determined as for number. Analyzing the NP sub-tree that contains the pronoun, its gender is easily assigned based on the pronoun's gender:

- 'Masculine' for pronouns he, him, his, himself;
- 'Feminine' for pronouns she, her, herself;
- 'Neutral' for pronouns it, its, itself.

Assessment of Gender of Referents

The gender is assigned to a referent NP node using the following rules, ordered by decreasing priority. The gender for referent:

- Is 'neutral' if the child of the NP node is neither a proper noun nor a pronoun;
- Corresponds to a person-like generic noun relating to a profession or to a parenthood;
- Corresponds to the gender of the pronoun if the NP node has one as child;
- Corresponds to the gender of the proper noun if the NP node has one as child.

The gender for proper nouns is recognized based on the Machine Learning approach described later in this section.

Let us show an example of this procedure on text "John and Mary read the book. They liked it, but he was happier than she was.", whose parse tree is shown in Figure 4.

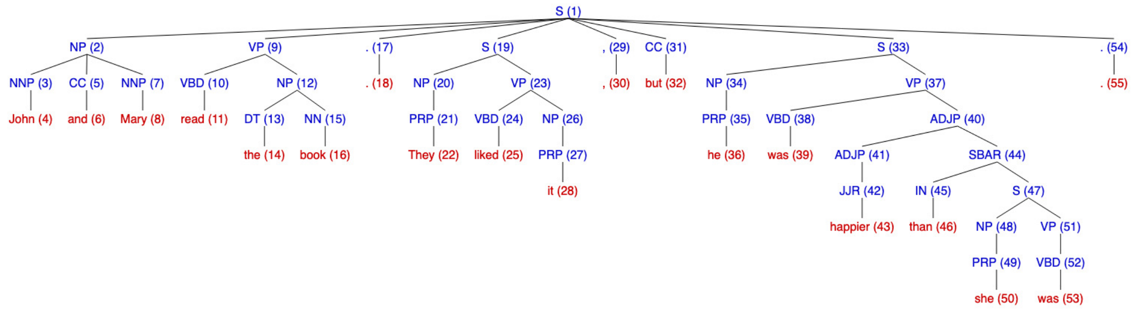

Figure 4. Sample text for gender and number agreement.

- Pronoun ‘They’ (22)
  - Number of Anaphora: plural
  - Number of Referents: NP (2) “John and Mary” is plural; NNP (3) “John” is singular; NNP (7) “Mary” is singular; NP (12) “book” is singular.
  - Gender of Singular Anaphora: N/A
  - Gender of Referents: N/A

  Thus, the only valid candidate is NP (2) “John and Mary” (plural).
- Pronoun ‘it’ (28)
  - Number of Anaphora: singular
  - Number of Referents: NP (2) “John and Mary” is plural; NNP (3) “John” is singular; NNP (7) “Mary” is singular; NP (12) “book” is singular; NP (20) “They” is plural.
  - Gender of Singular Anaphora: neutral (“it (28)”)
  - Gender of Referents: NNP (3) “John” is masculine (based on the Machine Learning approach); NNP (7) “Mary” is feminine (based on the Machine Learning approach); NP (12) “book” is neutral.

  Thus, the only valid candidate is “book” (singular, neutral).
- Pronoun ‘he’ (36)
  - Number of Anaphora: singular
  - Number of Referents: NP (2) “John and Mary” is plural; NNP (3) “John” is singular; NNP (7) “Mary” is singular; NP (12) “book” is singular; NP (20) “They” is plural; NP (26) “it” is singular.
  - Gender of Singular Anaphora: masculine (“he (36)”)
  - Gender of Referents: NNP (3) “John” is masculine (based on the Machine Learning approach); NNP (7) “Mary” is feminine (based on the Machine Learning approach); NP (12) “book” is neutral; NP (26) “it” is neutral.

  Thus, the only valid candidate is “John” (singular, masculine).
- Pronoun ‘she’ (50)
  - Number of Anaphora: singular
  - Number of Referents: NP (2) “John and Mary” is plural; NNP (3) “John” is singular; NNP (7) “Mary” is singular; NP (12) “book” is singular; NP (20) “They” is plural; NP (26) “it” is singular; NP (34) “he” is singular.
  - Gender of Singular Anaphora: feminine (“she”)
  - Gender of Referents: NNP (3) “John” is masculine (based on the Machine Learning approach); NNP (7) “Mary” is feminine (based on the Machine Learning approach); NP (12) “book” is neutral; NP (26) “it” is neutral; NP (34) “he” is masculine.

  Thus, the only valid candidate is “Mary” (singular, feminine).

3.1.2. Improvement over Base Rules

As said, Hobbs quite successfully applied a single algorithm to all kinds of targeted pronoun, with remarkably good results. While his theory is correct, in practice the different types of pronouns occur in different ways in the text due to their nature, and follow different rules in natural language too. Based on this observation, we developed slight specializations of Hobbs' rules according to the kind of pronoun to be resolved. This resulted in three similar but different algorithms for the four types of anaphoras presented in Section 2 (subjective, objective, possessive and reflexive).

e.g., we observed in the sentences that both reflexive and possessive pronouns highly regard the recency of the referents with respect to the anaphoras when the algorithm is looking for intra-sentential referents. For this reason, our variant removes the intra-sentential constraint that prevents the NP nearest to S from being considered as a potential candidate. This amounts to the following change in Step 2 of Hobbs' algorithm for possessive anaphoric pronouns resolution:

2. Traverse left-to-right, breadth-first, all branches under X to the left of p. Propose as reference any NP…

Hobbs: …that has an NP or S between it and X.

Ours: …under X.

On the example about John taking his license in Figure 1 our algorithm works the same as Hobbs', as shown in Section 2.2.1.

On the other hand, since reflexive anaphoras are necessarily intra-sentential, we designed the following specific strategy for reflexive anaphoric pronouns resolution:

1. Starting at the NP node immediately dominating the pronoun.
2. REPEAT
   (a) Climb the tree up to the first NP or S. Call this X, and call the path p.
   (b) Traverse left-to-right, breadth-first, all branches in the subtree rooted in X to the left of p. Propose as a candidate referent any NP under X.
3. UNTIL X is not the highest S in the sentence.

Let us provide two examples (one per intra-sentential kind of anaphora) that this algorithm can successfully solve whereas Hobbs' naïve one cannot:

Reflexive "John found himself in a small laboratory programming."

Possessive "Every day the sun shines with its powerful sunbeams."

Whose parse trees are shown in Figures 5 and 6, respectively.

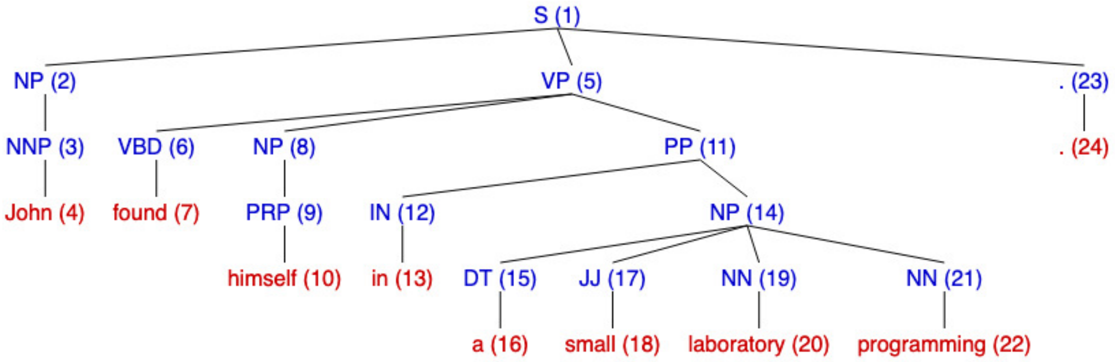

Figure 5. Sample text with reflexive anaphora.

Let us start from the reflexive example, with anaphora 'himself' (10):

1. The NP node immediately dominating the pronoun is (8).

2(a). Climbing up, the first NP or S is node S (1) with $p = \langle(1),(5),(8),(9),(10)\rangle$.

2(b). The left-to-right, breadth-first, traversal of all branches under S (1) to the left of p, involves nodes $\langle(2),(3)\rangle$ with only one candidate, NP (2).

3. X is already the highest S in the sentence: stop.

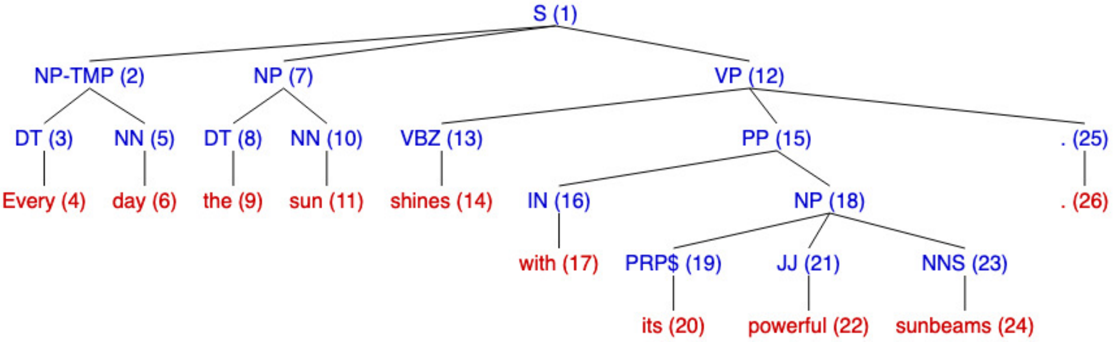

Figure 6. Sample text with possessive anaphora.

Let us now turn to the possessive example, with anaphora 'its' (20):

1. The NP node immediately dominating the pronoun is (18). Climbing up, the first NP or S is node S (1) with $p = \langle(1),(12),(15),(18),(19),(20)\rangle$.
2. The left-to-right, breadth-first, traversal of all branches under S (1) to the left of p, involves nodes $\langle(2),(7),(3),(5),(8),(10)\rangle$ with two candidates: NP-TMP (2) and NP (7).
3. (1) is the highest S node in sentence, but there is no previous sentence.
4. N/A
5. N/A
6. N/A
7. No branches of X to the right of p.
8. N/A

This experience shows the importance of adopting a rule-based approach over sub-symbolic ones: one may understand the faulty or missing part of the AR strategy and make for them by modifying or adding parts of the strategy.

3.2. Gender Recognition

Recognizing the gender of names is relevant in the context of AR because it can improve the resolution of masculine or feminine pronouns by excluding some wrong associations. Whilst for common nouns a vocabulary might do the job, the task is more complex when the referent is a proper noun. One way for endowing our approach with gender prediction capabilities on proper nouns would be using online services that, queried with a proper noun, return its gender. However, such services are usually non-free and would require an external connection. For this reason, we turned to the use of a local model obtained through Machine Learning. This section describes our initial attempt at learning gender models for people's names through the fixed length suffix approach. Suffixes are a rather good indicator of gender in proper nouns. Indeed, their use as features has been already considered in the literature [34], yielding quite good performance. e.g., Italian names that end in '-a' most probably refer to women, while names that end in '-o' are usually for males. Of course, in general (e.g., in English) the task is much more complex, justifying the use of Machine Learning to extract non-obvious regularities that are predictive of the name gender.

For this reason, we decided to investigate the predictiveness of suffixes in proper nouns to determine their gender. We tried an approach using a fixed suffix length . Specifically, we tried suffixes of 1, 2 or 3 characters. Longer suffixes were not considered to avoid potential overfitting. Using the letters in the suffix as features, we considered different machine learning approaches:

- Logistic regression as a baseline, since it is the simplest classifier to be tested on a classification task, and it is commonly used in the field of NLP;
- Decision trees, that we considered an interesting option because the last n characters ($n = 1, 2, 3$) of the name that we used as features would become tests in the learned tree, that in this way would be interpretable by humans;
- Random forests, an ensemble learning method that might improve the performance of decision trees, and especially avoid overfitting, by building multiple decision trees for the same set of target classes.

In the feature extraction step, for names made up of less characters than the length of the required suffix (e.g., 'Ed' when extracting suffixes of length 3), the missing characters were replaced by blank spaces.

4. Implementation and Experimental Results

The GEARS system has been implemented using different languages for different components. The core rule-based algorithm was implemented in Prolog, and specifically SWI Prolog (https://www.swi-prolog.org/, accessed on 10 November 2021). The machine learning algorithms for gender prediction exploited Python, and specifically the libraries Natural Language ToolKit (NLTK, https://www.nltk.org/, accessed on 10 November 2021) and Scikit-learn (https://scikit-learn.org/, accessed on 10 November 2021). All the parameters for the algorithm are specified in a suitable file. The main structure of the system, and the various pre- and post-processing modules, were implemented in Java, and used several libraries, including JPL (to interface Java to Prolog), and the CoreNLP library (along with its models package) . The latter carries out PoS tagging and syntactic analysis using the Stanford parser (https://nlp.stanford.edu/software/lex-parser.shtml, accessed on 10 November 2021), a tool with state-of-the-art performance. In this section, we will experimentally evaluate the effectiveness and efficiency of different aspects of our proposed approach, explain our experimental settings and discuss the outcomes.

A first and most relevant evaluation concerned the effectiveness of our proposal. It was assessed and compared to both

- Hobbs' algorithm, as the most basic approach in the literature, taken as a baseline in most research works, to understand how much each proposed improvement may improve the overall performance; and
- Liang and Wu's approach, as one of the latest contributions in that field, representing the state-of-the-art, to understand and possibly get indications on how the algorithm can be further improved.

Two additional experiments evaluated the efficiency of the GEARS System, by analyzing runtime for each processing phase during the computation, and the gender prediction task, by comparing the different machine learning algorithms applied to different features.

4.1. Gender Prediction

We start by discussing the experiments on gender prediction, both because gender models are learned off-line and before the AR computation takes place, and because the performance in this task obviously affects the performance of actual AR.

To evaluate the approach based on fixed length proper noun suffixes described in Section 3.2, we started from two sets of names, one per gender, and we extracted the features for each name to obtain a workable dataset. Then, we merged and shuffled the set of examples, and ran a 5-fold cross-validation procedure. It is a common setting in the literature, and in our case it allows to have sufficient data in the test set at each run of the experiment. We generated the folds once, and used them for all the machine learning algorithms, to avoid biases associated with the use of different training sets. After applying each algorithm to the folds, we collected the outcomes and computed their average performance. As said, we considered 3 machine learning approaches: logistic regression, decision trees and random forests.

The use of a Machine Learning approach required a training dataset of English proper nouns labeled with the corresponding gender. A free and reliable dataset for this purpose was the NLTK Corpus 'names' (https://www.nltk.org/nltk_data/, accessed on 10 November 2021). It is very simple and includes nearly 8000 names (2943 male names and 5001 female names). 365 ambiguous names occur in both lists. Whilst its quality is high, the number of entries is rather low, resulting in weak models in some preliminary experiments. So, we decided to expand it with more names. A larger dataset, freely distributed by the US government, is the database of 'popular baby names' (https://www.ssa.gov/oact/babynames/, accessed on 10 November 2021) available from the Social Security Agency. It provides information regarding not only the most popular baby names but also the history of all the names that have been given to babies in the US in the years 1880–2018, ordered by naming trends per year. Data are provided at three levels of granularity: national, state-specific or territory-specific. For the sake of generality, we opted for the national level. The data for each year are in a separate comma-separated values (csv) file including 3 fields: the name, the sex assigned to the name and the number of occurrences of that name with that sex in the selected year, ranked by occurrences. Only names that have at least 5 occurrences in the baby population of that year are reported. We neglected the information about the occurrences and the year. The merger of these two corpora of names included a total of more than 11,000 names, of which 4000+ male names and 7000+ female names.

Regarding the effectiveness of gender prediction, we measured performance using Accuracy:

$$A = \frac{TP + TN}{TP + TN + FP + FN}$$

that is a standard metric for the evaluation of machine learning algorithms. The results for different fixed-length suffixes and machine learning algorithms are shown in Table 2. First of all, we note that all machine learning approaches take advantage from the use of longer prefixes as features, except logistic regression, where performance using 3-character suffixes is lower than performance using 2-character ones. However, performance of logistic regression is about 10% lower than the other (tree-based) approaches, and thus we immediately discarded this algorithm as a candidate for use in our AR approach. This is not surprising, since decision trees are known for yielding good performance in tasks involving text (and indeed Logistic Regression was included in the comparison just to provide a baseline). Still, they often tend to overfit the dataset. Random forests are a variant that is commonly used to avoid this problem, leveraging its ensemble approach that learns a set of trees and combines their outcomes. However, in our case, random forests obtained the same performance as decision trees: they differ only in the second decimal digit. Since their performance is the same, but random forests are more complex models than decision trees, we opted for using the latter in our AR approach.

Table 2. Accuracy of gender prediction for different algorithms and features on the NLTK + SSA GOV dataset.

Algorithm	**1 char**	**2 chars**	**3 chars**
Logistic Regression	66%	68.5%	67.9%
Decision Trees	75.5%	78.5%	80.8%
Random Forests	75.5%	78.5%	80.8%

More specifically, we used the decision tree with highest accuracy among the 5 learned in the 5-fold cross-validation. We did not learn a new model using the entire training set, to prevent the additional examples from introducing overfitting. So, our AR approach may assume that proper noun gender recognition accuracy is around 80%, while this means that

1 gender every 5 names is misrecognized on average, still it is quite high a performance, that should positively affect the performance of the AR task, as a consequence.

4.2. Anaphora Resolution Effectiveness and Efficiency

Moving to the evaluation of our overall AR approach, the choice of a dataset was the first step to carry out. Based on the considerations in Section 2.3, we opted for the Brown Corpus [33], since it is freely available and was used by many previous relevant works, including Hobbs' [18] and Liang and Wu's [16] (differently from Liang and Wu, Hobbs uses many sources, including part of this dataset). Since the corpus is quite large, we selected 3 out of its 15 genres: two from the informative prose section (editorial press texts and popular lore editorials) and one from the imaginative prose section (science fiction novels). Whilst our main purpose is to address informative prose, we also tried our approach on imaginative prose, which is challenging since its nature and style may severely affect gender prediction. Table 3 reports some statistics about the selected subset. The most influential subset for our experiments is lore, since it includes the largest number of words and sentences.

Table 3. Statistics for the selection of Brown Corpus used in our experiments.

	Science Fiction	**Editorial**	**Lore**
# texts	6	27	48
sub-genres	novels (3), short stories (4)	institutional (10), personal (10), letters to editor (7)	books (23), periodicals (25),
# sentences	∼1000	∼3000	∼5000
# words	∼10,000	∼50,000	∼100,000
# pronouns	∼1000	∼2000	∼4000
# anaphoric	∼800 (80%)	∼1500 (75%)	∼3600 (90%)

For the window size, we used 4, which turned out to be the best to retrieve referents based on various experiments we carried out. Indeed, whilst Hobbs [18] and Lappin and Leass [12] considered windows of size 3, experimentally we found that many anaphora had no reference within 3 sentences, especially in dramatic prose. On the other hand, no significant improvement in performance was obtained for window size larger than 4.

For both experiments aimed at evaluating the effectiveness of GEARS on the AR task we adopted the Hobbs' metric, because it is the most widely exploited for rule-based AR systems in the literature, including Hobbs' and Liang and Wu's work, to which we compare our proposal. More specifically, when comparing the system's responses to the ground truth, each anaphora was associated with one of the following values: 'not found', if the anaphora in the ground truth was not found by the system; 'wrong', if the anaphora was found by the system but associated with a wrong reference; 'wrong sentence', if the anaphora was found by the system and associated with the correct referent, but in a different sentence than the ground truth; 'correct', if the anaphora-referent pair returned by the system is correct and the latter is found in the correct sentence.

The former experiment on AR effectiveness is an ablation study that compared the original algorithm by Hobbs to various combinations of our improvements, to assess the contribution that each brings to the overall performance. It aimed at answering the following research questions:

Q1 Can (our approach to) proper noun gender recognition, without the use of any vocabulary, bring significant improvement to the overall performance?

Q2 Can our modification to the basic algorithm by Hobbs improve performance, while still avoiding the use of any kind of external resource?

Its results, by genre, are reported in Table 4. The modification of the rules (Hobbs+) actually brings only a slight improvement (around 2%) over the original algorithm. Still, this is more or less the same improvement brought by Hobbs himself with his more complex approach based on selectional constraints, while we still use the sentence structure only. So, we may answer positively to question **Q2**. Given this result, we propose our rule-based algorithm as the new baseline to be considered by the literature on AR. Much more significant is the improvement given by the application of gender and number agreement (GN), since it boosts the performance of up to 21.13% (+60% on the new baseline) in the best case (Science Fiction), and of 14.06% (+36%) and 11.37% (+28%) in the other cases, which is still remarkably good. So, we may definitely answer positively question **Q1** and use the Hobbs + GN version in our next experiments.

Table 4. Comparison of Hobbs' original algorithm to our improvements (Hobbs' metric).

	Editorial	**Lore**	**Science Fiction**
Hobbs	0.38	0.37	0.32
Hobbs+	0.40	0.39	0.35
Hobbs+GN	0.52	0.53	0.56

The second experiment involves the comparison between GEARS and Liang and Wu's approach. Both systems use the Brown Corpus for the experimentation, but with slight differences , shown in the first row of Table 5 along with the results they obtained on the AR task. In this case, our research question is

Q3 How does the performance obtained using our improvements to Hobbs' algorithm, while still being a knowledge-poor approach, compare to a knowledge-rich state-of-the-art system?

Table 5. Comparison of our algorithm against Liang and Wu's (Hobbs' metric).

	GEARS	**Liang and Wu's**
Dataset	All texts for 3 genres	Random texts for all genres
# solved pronouns	6012	530
Editorial	52%	80%
Science-fiction	56%	79%
Lore	53%	69%
avg success rate	53%	77%

Whilst, as expected, Liang and Wu's system obtains better results, our results are worth appreciation, especially considering that GEARS solved 11+ times more pronouns than its competitor, which obviously increased the chances of failures due to peculiar cases. Furthermore, their experiment was carried out on random samples of texts for all the genres, while GEARS has been intensively tested on all the texts associated with the three selected genres.

For the efficiency evaluation of GEARS, the average runtime of each operation per document is shown in Table 6, obtained on a PC endowed with an Intel Core i5-680 @ 3.59GHz CPU running the Linux Ubuntu Server 14.04 x64 Operating System with 16 GB RAM. We observe that the most time-consuming activities are the PoS tagging of the text, carried out by the Stanford parser, and the execution of the AR algorithm. The latter requires equal or less (in one case half) time than the former, and thus the actual AR execution is faster than its preprocessing step.

Table 6. Average efficiency per document in milliseconds for the different operations in GEARS.

OPERATIONS	Editorial	Lore	Science Fiction
Text Reading	∼0 ms	∼0 ms	2 ms
POS Tag Annotation	8855 ms	11,248 ms	8475 ms
Script Generation	161 ms	176 ms	190 ms
Anaphora Resolution	4629 ms	11,414 ms	7599 ms
Output Writing	57 ms	65 ms	45 ms
Output Evaluation	∼0 ms	1 ms	2 ms

5. Conclusions

Anaphora Resolution, i.e., the task of resolving references to other items in a discourse, is a crucial activity for correctly and effectively processing texts in information extraction activities. Whilst generally rule-based, the approaches proposed in the literature for this task can be divided into syntax-based or discourse-based on one hand, and into knowledge-rich and knowledge-poor ones on the other. Knowledge-rich approaches try to improve performance by leveraging the information in external resources, which poses the problem of obtaining such resources (which are not always available, or not always of good quality, especially for languages different than English).

This paper proposed a knowledge-poor, syntax-based approach for anaphora resolution on English texts. Starting from an existing algorithm that is still regarded as the baseline for comparison by all works in the literature, our proposal tries to improve its performance in 2 respects: handling differently different kinds of anaphoras, and disambiguating alternate associations using gender recognition on proper nouns. Our approach can work based only on the parse tree of the sentences in the text, except for a predictor of the gender of proper nouns, for which we propose a machine learning-based approach, so as to completely avoid the use of external resources. Experimental results on a standard benchmark dataset used in the literature show that our approach can significantly improve the performance over the standard baseline algorithm (by Hobbs) used in the literature. Whilst the most significant contribution is provided by the gender agreement feature, the modification to the general rules alone already yields an improvement, for which we propose to use our algorithm as the new baseline in the literature. Its performance is also acceptable if compared to the latest state-of-the-art algorithm (by Liang and Wu), that belongs to the knowledge-rich family and exploits much external information, especially considering that we ran more intensive experiments than those reported for the competitor. Interestingly, the accuracy of our gender prediction tool is high but can still be improved, with further expected benefit for the overall anaphora resolution performance. Among the strengths of our proposal is also efficiency: it can process even long texts in a few seconds, where more than half of the time is spent in pre-processing for obtaining the parse trees of the sentences.

As future work, we expect that further improvements may come from additional extensions of the rules, to handle more and different kinds of anaphoras, and from an improvement of the gender recognition model, based on larger or more representative training sets. Furthermore, versions of our approach for different languages, with different features as regards syntax and proper noun morphology, should be developed to confirm its generality.

Author Contributions: Investigation, S.F. and D.R.; Methodology, S.F.; Project administration, S.F. and D.R.; Resources, S.F. and D.R.; Software, D.R.; Supervision, S.F.; Validation, S.F. and D.R.; Writing—original draft, S.F. and D.R.; Writing—review and editing, S.F. and D.R. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The datasets used in this work were taken from repositories available on the Internet, and specifically: Brown Corpus (http://icame.uib.no/brown/bcm.html, accessed on 25 January 2022); NLTK Corpus 'names' (https://www.nltk.org/nltk_data/, accessed on 25 January 2022); US government Social Security Agency 'popular baby names' (https://www.ssa.gov/oact/babynames/, accessed on 25 January 2022) . The code of the algorithm will be made available upon request to the authors.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AR	Anaphora Resolution
ER	Entity Resolution
PoS	Part-of-Speech

References

1. Rotella, F.; Leuzzi, F.; Ferilli, S. Learning and exploiting concept networks with ConNeKTion. *Appl. Intell.* **2015**, *42*, 87–111. [CrossRef]
2. Ferilli, S.; Redavid, D. The GraphBRAIN System for Knowledge Graph Management and Advanced Fruition. In Proceedings of the Foundations of Intelligent Systems—25th International Symposium, ISMIS 2020, Graz, Austria, 23–25 September 2020; Lecture Notes in Computer Science; Helic, D., Leitner, G., Stettinger, M., Felfernig, A., Ras, Z.W., Eds.; Springer: Berlin/Heidelberg, Germany, 2020; Volume 12117, pp. 308–317.
3. Ferilli, S. Integration Strategy and Tool between Formal Ontology and Graph Database Technology. *Electronics* **2021**, *10*, 2616. [CrossRef]
4. Ferilli, S.; Redavid, D. An ontology and a collaborative knowledge base for history of computing. In Proceedings of the First International Workshop on Open Data and Ontologies for Cultural Heritage Co-Located with the 31st International Conference on Advanced Information Systems Engineering, ODOCH@CAiSE 2019, Rome, Italy, 3 June 2019; CEUR Workshop Proceedings; Poggi, A., Ed.; CEUR-WS.org: Aachen, Germany, 2019; Volume 2375, pp. 49–60.
5. Getoor, L.; Machanavajjhala, A. Entity resolution for big data. In Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'13), Chicago, IL, USA, 11–14 August 2013; Association for Computing Machinery: New York, NY, USA, 2013. [CrossRef]
6. Mitkov, R. *Anaphora Resolution: The State of the Art*; Research Report (Research Group in Computational Linguistics and Language Engineering); School of Languages and European Studies, University of Wolverhampton: Wolverhampton, UK, 1999.
7. Seddik, K.M.; Farghaly, A. Anaphora Resolution. In *Natural Language Processing of Semitic Languages*; Zitouni, I., Ed.; Theory and Applications of Natural Language Processing; Springer: Berlin/Heidelberg, Germany, 2014; pp. 247–277. [CrossRef]
8. Mitkov, R.; Evans, R.; Orasan, C. A New, Fully Automatic Version of Mitkov's Knowledge-Poor Pronoun Resolution Method. In Proceedings of the Computational Linguistics and Intelligent Text Processing, Third International Conference, CICLing 2002, Mexico City, Mexico, 17–23 February 2002; Lecture Notes in Computer Science; Gelbukh, A.F., Ed.; Springer: Berlin/Heidelberg, Germany, 2002; Volume 2276, pp. 168–186. [CrossRef]
9. Group, T.S.N. Coreference Resolution. Available online: https://nlp.stanford.edu/projects/coref.shtml (accessed on 10 November 2021).
10. Elango, P. *Coreference Resolution: A Survey*; University of Wisconsin: Madison, WI, USA, 2005.
11. Sukthanker, R.; Poria, S.; Cambria, E.; Thirunavukarasu, R. Anaphora and coreference resolution: A review. *Inf. Fusion* **2020**, *59*, 139–162. [CrossRef]
12. Lappin, S.; Leass, H.J. An Algorithm for Pronominal Anaphora Resolution. *Comput. Linguist.* **1994**, *20*, 535–561.
13. Franza, T. An Improved Anaphora Resolution Strategy Based on Text Structure and Inductive Reasoning. Master's Thesis, University of Bari, Bari, Italy, 2020.
14. Sayed, I.Q. *Issues in Anaphora Resolution;* Technical Report; USA, 2003. Available online: https://nlp.stanford.edu/courses/cs224n/2003/fp/iqsayed/project_report.pdf (accessed on 25 January 2022).
15. Mitkov, R. Outstanding Issues in Anaphora Resolution (Invited Talk). In Proceedings of the Computational Linguistics and Intelligent Text Processing, Second International Conference, CICLing 2001, Mexico City, Mexico, 18–24 February 2001; Lecture Notes in Computer Science; Gelbukh, A.F., Ed.; Springer: Berlin/Heidelberg, Germany, 2001, Volume 2004, pp. 110–125. [CrossRef]
16. Liang, T.; Wu, D.S. Automatic Pronominal Anaphora Resolution in English Texts. *Int. J. Comput. Linguist. Chin. Lang. Process.* **2004**, *9*, 21–40.
17. Mitkov, R. *The Oxford Handbook of Computational Linguistics (Oxford Handbooks)*; Oxford University Press, Inc.: New York, NY, USA, 2005.

18. Hobbs, J. Resolving Pronoun References. In *Readings in Natural Language Processing*; Morgan Kaufmann Publishers Inc.: San Francisco, CA, USA, 1986; pp. 339–352.
19. Grosz, B.; Joshi, A.; Weinstein, S. *Centering: A Framework for Modelling the Coherence of Discourse*; Technical Reports; Department of Computer & Information Science, University of Pennsylvania: Philadelphia, PA, USA, 1994.
20. Ferilli, S.; Esposito, F.; Grieco, D. Automatic Learning of Linguistic Resources for Stopword Removal and Stemming from Text. *Procedia Comput. Sci.* **2014**, *38*, 116–123. [CrossRef]
21. Harabagiu, S.M.; Maiorano, S.J. Knowledge-Lean Coreference Resolution and its Relation to Textual Cohesion and Coherence. In Proceedings of the ACL Workshop on Discourse/Dialogue Structure and Reference, University of Maryland, College Park, MD, USA, 1999; pp. 29–38.
22. Miller, G.A. WordNet: A Lexical Database for English. *Commun. ACM* **1995**, *38*, 39–41. [CrossRef]
23. Baldwin, B. CogNIAC: High Precision Coreference with Limited Knowledge and Linguistic Resources. In Proceedings of a Workshop on Operational Factors in Practical, Robust Anaphora Resolution for Unrestricted Texts, Madrid, Spain, 11 July 1997; Association for Computational Linguistics: Stroudsburg, PA, USA, 1997; pp. 38–45.
24. Marcus, M.; Kim, G.; Marcinkiewicz, M.A.; MacIntyre, R.; Bies, A.; Ferguson, M.; Katz, K.; Schasberger, B. The Penn Treebank: Annotating Predicate Argument Structure. In Proceedings of the Workshop on Human Language Technology, HLT '94, Plainsboro NJ, USA, 8–11 March 1994; Association for Computational Linguistics: Stroudsburg, PA, USA, 1994; pp. 114–119. [CrossRef]
25. Walker, J.P.; Walker, M.I. *Centering Theory in Discourse*; Oxford University Press: Oxford, UK, 1998.
26. Doddington, G.; Mitchell, A.; Przybocki, M.; Ramshaw, L.; Strassel, S.; Weischedel, R. The Automatic Content Extraction (ACE) Program—Tasks, Data, and Evaluation. In Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04), Lisbon, Portugal, 26–28 May 2004; European Language Resources Association (ELRA): Paris, France, 2004.
27. Poesio, M.; Artstein, R. Anaphoric Annotation in the ARRAU Corpus. In Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08), Marrakech, Morocco, 28–30 May 2008; European Language Resources Association (ELRA): Paris, France, 2008; pp. 1170–1174.
28. Gross, D.; Allen, J.F.; Traum, D.R. *The Trains 91 Dialogues*; Technical Report; University of Rochester: Rochester, NY, USA, 1993.
29. Heeman, P.A.; Allen, J.F. *The Trains 93 Dialogues*; Technical Report; University of Rochester: Rochester, NY, USA, 1995.
30. Watson-Gegeo, K.A.; Wallace L. The pear stories: Cognitive, cultural, and linguistic aspects of narrative production (Advances in Discourse Processes, vol. III). *Lang. Soc.* **1981**, *10*, 451–453. [CrossRef]
31. Carlson, L.; Marcu, D.; Okurowski, M.E., Building a Discourse-Tagged Corpus in the Framework of Rhetorical Structure Theory. In *Current and New Directions in Discourse and Dialogue*; van Kuppevelt, J., Smith, R.W., Eds.; Springer: Dordrecht, The Netherlands, 2003; pp. 85–112. [CrossRef]
32. Poesio, M. Discourse Annotation and Semantic Annotation in the GNOME Corpus. In Proceedings of the 2004 ACL Workshop on Discourse Annotation, DiscAnnotation '04, Barcelona, Spain, 25–26 July 2004; Association for Computational Linguistics: Stroudsburg, PA, USA, 2004; pp. 72–79.
33. Francis, W.; Kučera, H. A Standard Corpus of Present-Day Edited American English, for use with Digital Computers (Brown), 1964, 1971, 1979. Brown University. Providence, Rhode Island. Available online: https://www.sketchengine.eu/brown-corpus/ (accessed 25 January 2022).
34. Wais, K. Gender Prediction Methods Based on First Names with genderize. *R J.* **2016**, *8*, 17–37. [CrossRef]