An Overview of Anaphora Resolution Algorithms

Computational Linguistics – 2
IIIT Hyderabad

Abhinav S
Menon

Outline

Reference & Anaphora

Popular Methods

Syntax-Based Algorithms

Hobbs (1978)
Lappin and Leass (1994)
Kennedy and Broguraev (1996)

Semantics-Based Algorithms

Hobbs (1978)
Carbonell and Brown (1988)
Nasukawa (1994)

Summary

Reference & Anaphora

Reference

Anaphora

Reference

When an expression indicates a specific entity, it is said to ***refer*** to it.

The entity is called its ***referent***.

If two expressions refer to the same entity, they are ***coreferent***.

Anaphora

Anaphora usually describes a certain expression referring to the same entity as another RE in the discourse.

Anaphora is a special case of ***coreference***.

Anaphora can be classified as forward/backward or inter-/intrasentential.

Popular Methods

Classification of Algorithms

Factors Used in Resolution

Classification of Algorithms

The algorithms we are going to see today fall into two types:

Syntax-Based Algorithms:

- Hobbs (1978)
- Lappin and Leass (1994)
- Kennedy and Broguraev (1996)

Semantics-Based Algorithms:

- Hobbs (1978)
- Carbonell and Brown (1988)
- Nasukawa (1994)

Factors Used in Resolution

These are some factors (besides syntactic and semantic information) used to resolve anaphora.

Collocation	Position	Emphasis
or	i.e.	through
parallelism	proximity	topicalisation
or		or
persistence		frequency

Syntax-Based Algorithms

Hobbs (1978)
Naïve search through parse tree

Lappin and Leass (1994)
Salience factors

Kennedy and Broguraev (1996)
PoS tags

Hobbs (1978)

Working

- Starting from the NP immediately above the pronoun, go up to the first NP or S node encountered.
  Traverse its children using BFS. Propose.
- Go further up to the next NP or S.
  Propose this if possible; else traverse its children using BFS. Propose.
- Repeat until one sentence is completed; then go to the previous sentence.

Constraints

morphological (GNP)

syntactic (non-coreferentiality)

semantic (selectional)

Example Run

The castle in Camelot remained the residence of the kind, until 536 when he moved it to London.

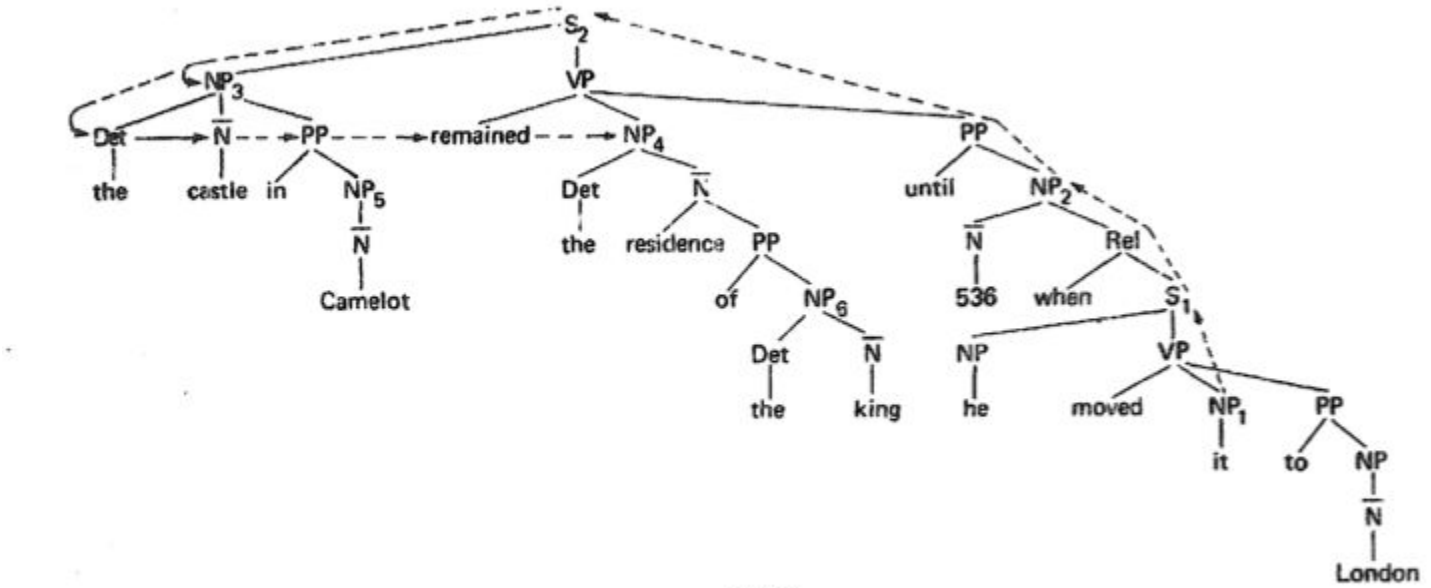


Fig. 2.

Results

This algorithm accounts for missing antecedents.

It does *not* account for sentence pronominalisation.

It worked overall in 88.3% of cases.

Augmented by selectional constraints, it rose to 91.7%.

Syntax-Based Algorithms

Hobbs (1978)
Naïve search through parse tree

Lappin and Leass (1994)
Salience factors

Kennedy and Broguraev (1996)
PoS tags

Lappin & Leass (1994)

Components

- Filters:
  - intrasentential syntactic filter
  - morphological filter
- Edge case procedures
  - pleonastic pronouns
  - lexical anaphors
- Main procedures
  - salience assignment
  - decision

Background

- Implemented in English and German
- Uses slot grammar parse
- Integrated into MT

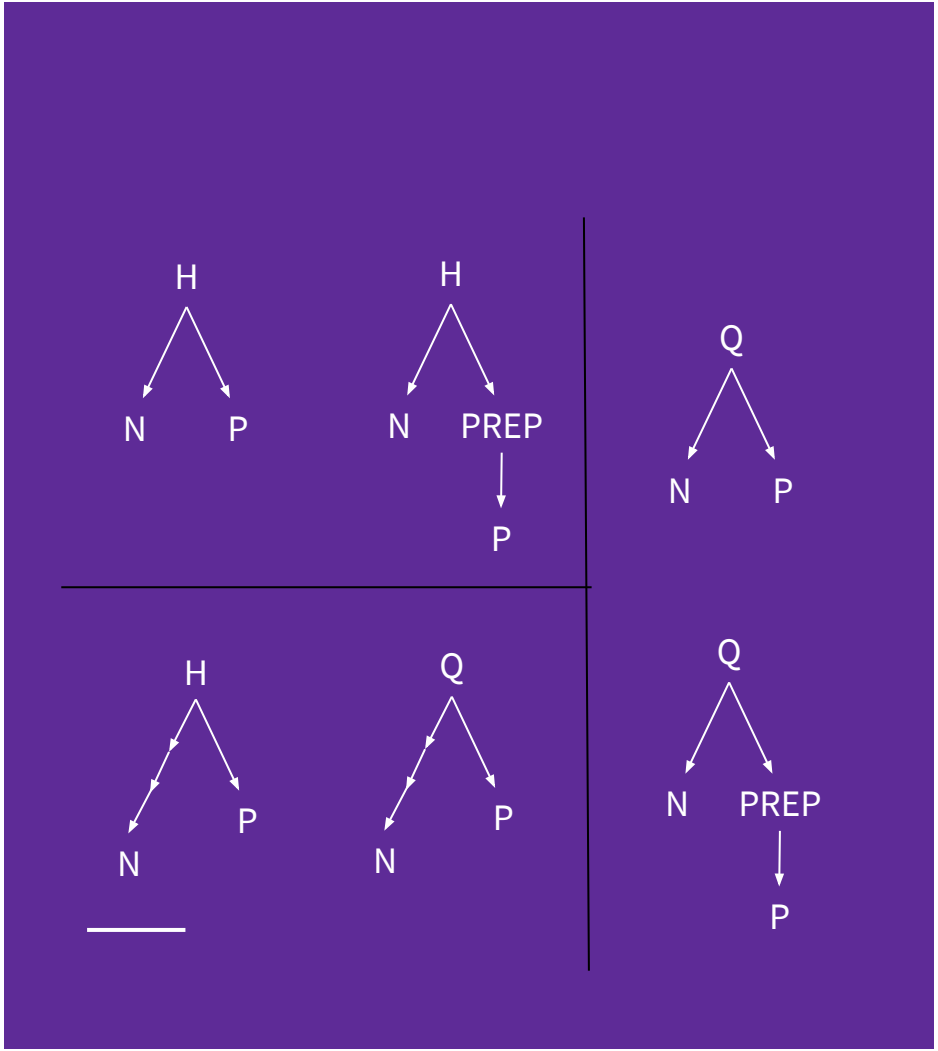
Filters

Intrasentential syntactic filter

- Filters out lexical anaphors
- Six syntactic conditions →

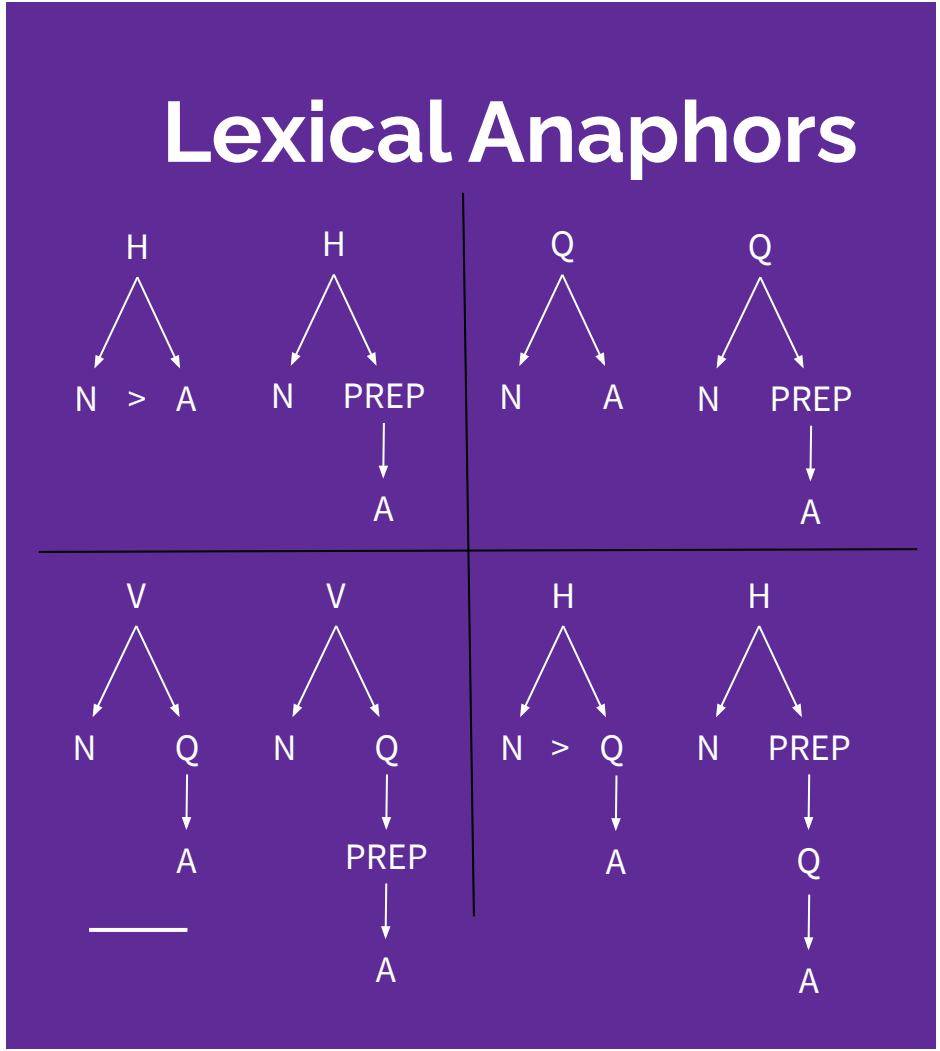
Morphological filter

- GNP constraints



Pleonastic Pronouns

It is **Modaladj** that **S**
It is **Modaladj** for **NP** to **VP**
It is **Cogv-**ed that **S**
It seems/appears/means/follows that **S**
NP makes/finds it **Modaladj** (for **NP)** to **VP**
It is time to **VP**
It is thanks to **NP** that **S**



Salience Assignment

- There are a number of *salience factors*.
- They add a certain *salience weight* to REs in their scope.

Factor type	Initial weight
Sentence recency	100
Subject emphasis	80
Existential emphasis	70
Accusative emphasis	50
Indirect object and oblique complement emphasis	40
Head noun emphasis	80
Non-adverbial emphasis	50

- Cataphora is penalised and parallelism is rewarded.

Selection

- The transitive, reflexive and symmetric closure of the antecedence relation defines the coreference relation.
- The equivalence classes then have salience weights.
- Choose the class with the highest salience.
  Resolve ties with proximity.

Example Run

You have not waited for the file to close. You may have asked to print on the virtual printer, but **it** cannot print until the output file is closed.

```
Non-coreferential pronoun--NP pairs:

you.1 - printer.10, you.1 - it.13, you.1 - output.19,
you.1 - file.20, it.13 - you.1, it.13 - output.19,
it.13 - file.20

Salience values:
  printer.(2.10) - 270
  file.(1.7) - 190

Salience factor values:
  printer.(2.10)                file.(1.7)
    sentence_rec - 100          sentence_rec - 50
    non_adverbial_emph - 50     non_adverbial_emph - 25
    pobj_emph - 40              subj_emph - 40
    head_emph - 80              head_emph - 40

Local salience factor values:
  file.(1.7)
    parallel_roles_reward - 35

Anaphor--Antecedent links:

  it.(2.13) to printer.(2.10)
```

Results

This algorithm accounts for many edge cases and needs no semantic info.

It does *not* account for nesting and the salience may not be accurate.

It worked overall in 85% of cases.

A statistical extension, RAPSTAT, improves this to 89%.

Syntax-Based Algorithms

Hobbs (1978)
Naïve search through parse tree

Lappin and Leass (1994)
Salience factors

Kennedy and Broguraev (1996)
PoS tags

Kennedy & Broguraev (1996)

Outline

- The input is a PoS-tagged, function-annotated and feature-marked sentence.
- The NPs, their contexts and pleonastic (expletive) *it*s are identified.
- A set of discourse referents is identified. From here, it is similar to RAP:
  - lexical anaphors
  - filtering and salience
  - selection

Motivation

- Lappin and Leass (1994) make impractical assumptions.
- PoS and GFUN tags are easy to obtain.

Lexical Anaphors

- It is assumed that a lexical anaphor's antecedent must be a coargument.
- Coarguments are identified using GFUN information.

If the anaphor is	Then the antecedent is
● a direct object	● the closest subject
● an indirect object or an oblique	● the closest subject or direct object

Filtering

The set of possible referents is filtered by

- morphological criteria (GNP agreement)
- syntactic criteria: a pronoun cannot corefer with
  - a coargument
  - a constituent which it commands (precedes)
  - a constituent which contains it

Coarguments are identified as for lexical anaphors.

Salience

- Salience is assigned in a manner closely following RAP.

SENT-S:	100 iff in the current sentence
CNTX-S:	50 iff in the current context
SUBJ-S:	80 iff GFUN = *subject*
EXST-S:	70 iff in an existential construction
POSS-S:	65 iff GFUN = *possessive*
ACC-S:	50 iff GFUN = *direct object*
DAT-S:	40 iff GFUN = *indirect object*
OBLQ-S:	30 iff the complement of a preposition
HEAD-S:	80 iff EMBED = NIL
ARG-S:	50 iff ADJUNCT = NIL

- Cataphora is penalised (as before) and locality and parallelism are *both* rewarded.

Selection

- The class with the highest salience is chosen as the antecedent.
- Its salience is recalculated accordingly.

Salience Assignment

- There are a number of *salience factors*.
- They add a certain *salience weight* to REs in their scope.

Factor type	Initial weight
Sentence recency	100
Subject emphasis	80
Existential emphasis	70
Accusative emphasis	50
Indirect object and oblique complement emphasis	40
Head noun emphasis	80
Non-adverbial emphasis	50

- Cataphora is penalised and parallelism is rewarded.

Selection

- The transitive, reflexive and symmetric closure of the antecedence relation defines the coreference relation.
- The equivalence classes then have salience weights.
- Choose the class with the highest salience.
  Resolve ties with proximity.

Results

This algorithm needs no semantic knowledge, like RAP.

It also needs considerably less syntactic input.

It worked overall in 75% of cases.

Semantics-Based Algorithms

Hobbs (1978)
A deductive system for semantic analysis

Carbonell and Brown (1988)
Multiple strategies

Nasukawa (1994)
A semantic salience method

Hobbs (1978)

Outline

- The algorithm is intended for semantic analysis, with resolution as a side-effect.
- Inferences are drawn from the lexicon as required.
- Four principal semantic operations are performed:
  - detecting connectives
  - predicate interpretation
  - knitting
  - entity identification

Logical System

- The input is text, reduced to logical notation.

$$on(X_1|boy(X_1), X_2|roof(X_2, X_3|building(X_3)))$$

  It is assumed that syntactic filters have been applied.

- A lexicon of world knowledge is kept.

$$\forall y : bank(y) \rightarrow building(y)$$

Detecting Connectives

- The relations among sentences are found by comparing their propositions.
- Three common patterns are:
  - Contrast: A and B
    - are contradictory, or
    - have one identical and one different arg
  - Violated Expectation
    - A → x and B → ~x for some x.
  - Cause: A → B

Predicate Interpretation

- Metaphorical usage leads to non-trivial problems of sense disambiguation.
- A predicate's definition has two parts:
  - the conditions on its arguments (inferred from their properties)
  - the modifications to be made on these inferences
- Consider `into` as an example.

Knitting

- When two statements with the same predicates, we guess they are redundant.
- If inconsistencies are not found, the arguments are unified.

Entity Identification

- An entity may remain unidentified even after knitting.
- Inferences that lead to known properties of it are then needed.
- A search for such inferences is conducted through the lexicon.

The boy walked into the bank. Moments later he was seen on **its** roof.

Example Run

They$_1$ prohibited them$_2$ from demonstrating because they$_1$ feared violence.

They$_1$ prohibited them$_2$ from demonstrating because they$_2$ advocated violence.

Axioms:

```
fears(X,Y) → not(wants(X,Y))
demonstrate(X) → violence(X)
```

Facts:

```
fears(X, violence) → not(wants(X,violence)
not(wants(X1, demonstrate(X2)))
```

Inferences:

```
demonstrate(X2) → violence
not(wants(X1,violence)
X = X1
```

Results

This algorithm assumes the existence of a system to generate the predicate notation, which is far from trivial.

The lexicon could be impractically large.

Semantics-Based Algorithms

Hobbs (1978)
A deductive system for semantic analysis

Carbonell and Brown (1988)
Multiple strategies

Nasukawa (1994)
A semantic salience method

Carbonell & Brown (1988)

Strategies

- Local Anaphor Constraints
  - morphological agreement
- Case-Role Semantic Constraints
- Pre-/Postcondition Constraints
  - necessary actions before/after

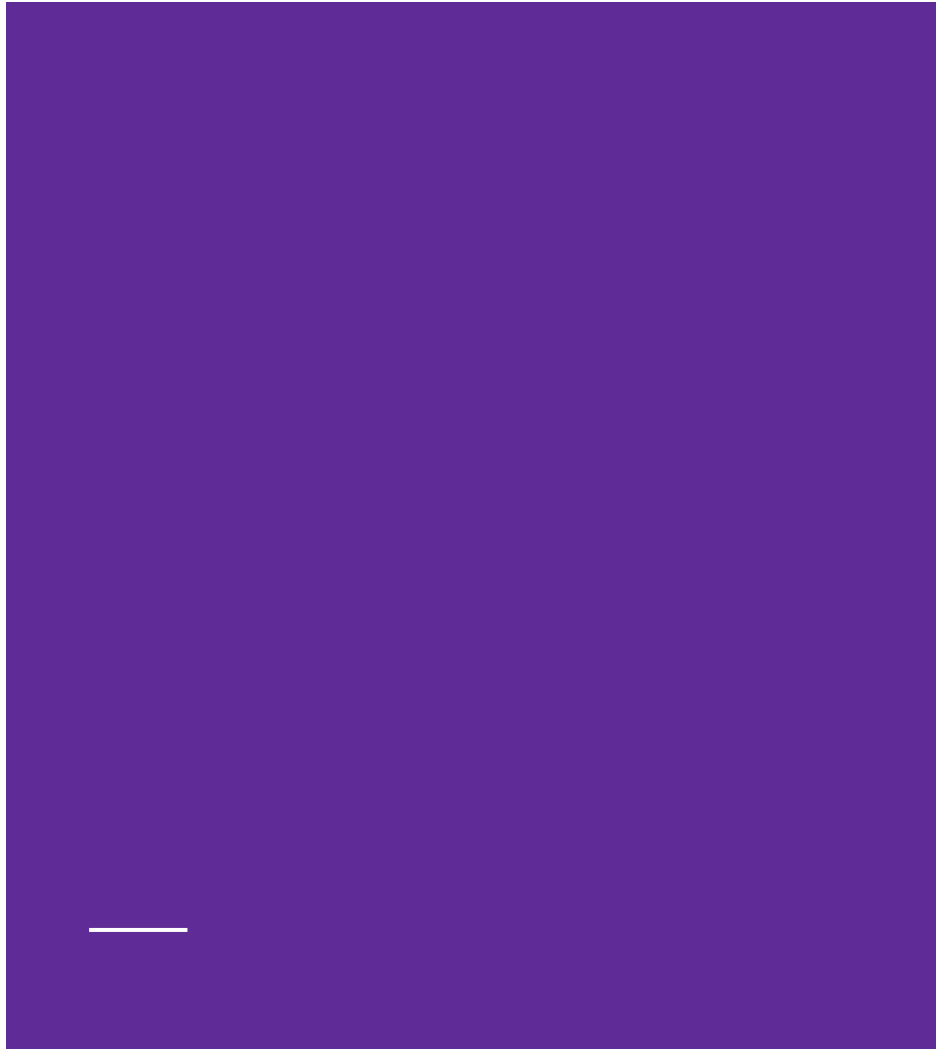
- Case-Role Persistence Preference
  - “linguistic inertia”
- Semantic Alignment Preference
  - partially lexical
- Syntactic Parallelism Preference
- Syntactic Topicalisation Preference
- Intersentential Recency Preference

Motivation

- Semantics and pragmatics dominate
- Human judgement of ambiguity

Working

- The constraints apply first, to filter out candidates.
- The preferences then follow a voting method to find the most preferred one.



Results

This algorithm has “cognitive plausibility” and appears to agree with human judgment.

No figures are provided for its computational accuracy. It has not been implemented with all strategies.

Semantics-Based Algorithms

Hobbs (1978)
A deductive system for semantic analysis

Carbonell and Brown (1988)
Multiple strategies

Nasukawa (1994)
A semantic salience method

Nasukawa (1994)

Outline

- Grammatical constraints are applied at an early stage.
- Three factors are used to measure salience:
  - collocation patterns
  - frequency of repetition
  - syntactic position

Basis

- If a text is consistent, it has rich information for resolving ambiguities.
- Fully semantic methods pose a knowledge acquisition bottleneck.

Collocation Patterns

- In order to apply proper semantic constraints, a large knowledge base is required.
- Collocation patterns are modifier-modifiee relationships.
- They indicate which words can fill which arguments of which verbs.
- It is assumed that word sense is unified within a discourse.

Frequency of Repetition

- An object in focus is likely to be pronominalised (as in centering and in Carbonell and Brown (1988)).
- The frequency of the lemma is taken as an index of its focus.

Syntactic Position

- Some syntactic positions are more likely to be pronominalised.
- This factor assigns a definite ranking to all NPs.

Implementation

- Extraction of candidates
  - two sentences
  - GNP, reflexivity filtering
- Selection of an antecedent
  - each factor’s weight is assigned

Example Run

(104) *All four of the cursor movement keys are typematic;*
(105) *they keep repeating as long as they are held down.*
(106) *The Cursor Up key moves the cursor up one line.*
(107) *Like the other cursor movement keys, this key moves the cursor one line or many lines depending on how long you hold down the key.*
(108) *The Cursor Right key moves the cursor to the right.*
(109) *Hold the key down.*
(110) *When the cursor reaches the right end of the line, it goes off the screen and reappears on the left side, one line below the line it was on.*
(111) *If the cursor is on the bottom line of the screen and is run all the way to the right, it goes off the screen and reappears in the upper left corner.*
(112) *The Cursor Left key moves the cursor one position to the left.*
(113) *Hold this key down.*
(114) *When* **it** *reaches the left end of the line, it goes off the screen and reappears on the right, one line above the line it was on.*

```
Candidates for the referent of CFRAME106579("it") are:
     CFRAME106573<113> .... key (0.48571432)
     CFRAME106564<112> .... Cursor Left key (0)
     CFRAME106565<112> .... cursor (1.4337664)
     CFRAME106568<112> .... left (0)

>> With DIANA <<
  >> To support CFRAME106565(cursor) <<
   1 with SAME-ATTACHMENT-CAND-MODIFIEE in:
     "  When the cursor reaches the right end of the line,
        it goes off the screen and reappears on the left side,
        one line below the line it was on ."
      (reaches<CFRAME106454> in SENTENCE106453<No.110>)
```

Results

This algorithm needs no semantic knowledge, but uses semantic information.

However, it needs annotation of syntactic roles.

It worked overall in 93.8% of cases.

Summary

- Reference and Anaphora
- Popular Methods
- Algorithms
  - Syntax-Based
  - Semantics-Based

References 😂

Stanford Encyclopedia of Philosophy, Anaphora

King, Jeffrey C. and Karen S. Lewis, "Anaphora", *The Stanford Encyclopedia of Philosophy* (Fall 2018 Edition), Edward N. Zalta (ed.), URL = https://plato.stanford.edu/archives/fall2018/entries/anaphora

Resolving Pronoun References [Hobbs (1978)]

Hobbs, Jerry R. "Resolving pronoun references." *Lingua* 44.4 (1978): 311-338.

An Algorithm for Pronominal Anaphora Resolution [Lappin and Leass (1994)]

Lappin, Shalom, and Herbert J. Leass. "An algorithm for pronominal anaphora resolution." *Computational linguistics* 20.4 (1994): 535-561.

Anaphora for Everyone: Pronominal Anaphora Resolution without a Parser [Kennedy and Broguraev (1996)]

Kennedy, Christopher, and Branimir Boguraev. "Anaphora for everyone: Pronominal anaphora resolution without a parser." *COLING 1996 Volume 1: The 16th International Conference on Computational Linguistics*. 1996.

Anaphora Resolution – A Multi-Strategy Approach [Carbonell and Brown (1988)]

Carbonell, Jaime G., and Ralf D. Brown. "Anaphora resolution: a multi-strategy approach." *Coling Budapest 1988 Volume 1: International Conference on Computational Linguistics*. 1988.

Robust Method for Pronoun Resolution using Full-Text Information [Nasukawa (1996)]

Nasukawa, Tetsuya. "Robust method of pronoun resolution using full-text information." *COLING 1994 Volume 2: The 15th International Conference on Computational Linguistics*. 1994.