

A Public Reference Implementation of the RAP Anaphora Resolution Algorithm



Long Qiu, Min-Yen Kan, Tat-Seng Chua

School of Computing
National University of Singapore
3 Science Drive 2
Singapore 117543
[qiu,kanmy,chuats]@comp.nus.edu.sg

Abstract

We present a standalone, publicly-available implementation of the Resolution of Anaphora Procedure (RAP) given by Lappin and Leass (1994). The RAP algorithm resolves third person pronouns, lexical anaphors, and identifies pleonastic pronouns. Our implementation, JavaRAP, fills a current need in anaphora resolution research by providing a reference implementation that can be benchmarked against current algorithms. The implementation uses the standard, publicly available Charniak (2000) parser as input, and generates a list of anaphora-antecedent pairs as output. Alternately, an in-place substitution of the anaphors with their antecedents can be produced. Evaluation on the MUC-6(Message Understanding Conferences) co-reference task shows that JavaRAP has an accuracy of 57.9%, similar to the performance given previously in the literature (e.g., Preiss 2002).

JavaRAP: A Public Reference RAP Implementation

•Language: **English**

•Input:

- Plain text
- Text with XML style tags
- Text with MUC co-reference annotations

•Output:

- Anaphor - antecedent pairs
- Text with in-place substitutions

•Accuracy: **57.9%**

•Efficiency:

- >1500 words / second (P2.4, 1G, excluding parsing)

• What is JavaRAP for?

- To benchmark other anaphora resolution algorithms;
- To provide anaphora resolution to NLP applications (automatic text summarization, Q&A, etc.).

• Programming Language: **Java**

• Software required:

- Java JDK 1.4 ++
- The Charniak parser (www.cs.brown.edu/people/ec/)

• Associated Tools:

- Sentence Splitter
 - Rule-Based
 - Flexible (supports different input/output formats)
- Anaphora Resolver Evaluator
 - Pair-wise comparison between resolution result and annotation (MUC6 co-reference convention)

• Availability: **Free and downloadable from**

www.comp.nus.edu.sg/~qiul/NLPTools

Resolution of Anaphora Procedure (Lappin and Leass' Algorithm)

- Pleonastic Pronoun Detection
- Morphological Filter (Number, People, Gender)
- Syntactic Filter (for third person pronouns)
- Anaphor Binding Algorithm (for lexical anaphors)
- Salience Measure

Anaphor Binding Algorithm for lexical anaphors

- Lexical Anaphors: include *reflexives* (itself, ourselves, etc.) and *reciprocals* (each other, one another,...).
- A lexical anaphor *A* is coreferential with a *NP* if any of the following conditions holds:
 - A* is in the argument domain of *N*, and *N* fills a higher argument slot than *A* ("They wanted to see themselves.");
 - A* is in the adjunct domain of *N* ("He worked by himself.");
 - A* is in the NP domain of *N* ("John likes Bill's portrait of himself.");
 - N* is an argument of a verb *V*. Meanwhile, there is a noun phrase *Q* in the argument domain or the adjunct domain of *N* such that *Q* has no noun determiner, and (i) *A* is an argument of *Q*, or (ii) *A* is an argument of a preposition *PREP* which is an adjunct of *Q* ("They told stories about themselves."); and
 - A* is a determiner of a noun *Q*, and (i) *Q* is in the argument domain of *N* and *N* fills a higher argument slot than *Q*, or (ii) *Q* is in the adjunct domain of *N* ("John and Marry like each other's portraits.").

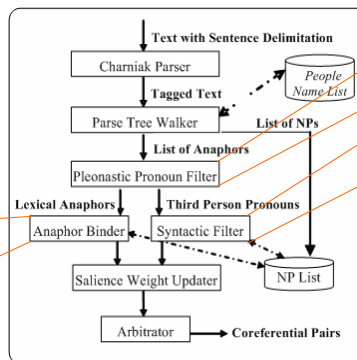


Figure 1. JavaRAP System Structure

Table 1: Salience Factors and their initial weights (as in RAP)

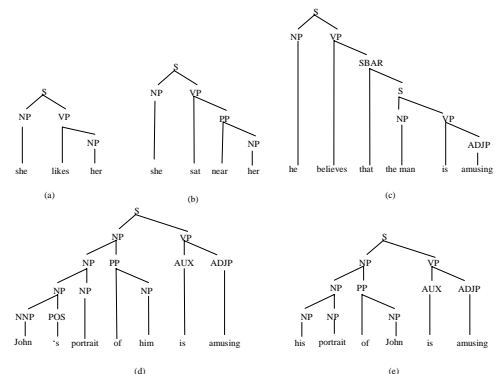
Factor	Initial Weight
Sentence Recency*	100
Subject Emphasis	80
Existential Emphasis	70
Accusative Emphasis	50
Indirect Object and Oblique Complement Emphasis	40
Head Noun Emphasis	80
Non-adverbial Emphasis	50

Pleonastic Pronoun Detection

- Pleonastic Pronouns: pronouns that have no referent.
- Detected by pattern matching:
 - it is *Modal_Adjective* that *S*
 - it is *Cognitive_Verb-ed* that *S*
 - it is *time* to *VP*
 - ...

Syntactic Filter for third personal pronouns

- A third person pronoun *P* is **NOT** coreferential with a *NP* if any of the following conditions holds:
 - P* is in the argument domain of *NP*;
 - P* is in the adjunct domain of *NP*;
 - P* is an argument of a head *H* and *NP* is contained in *H*;
 - P* is in the *NP* domain of *NP*;
 - P* is a determiner of a noun *Q* and *NP* is contained in *Q*.



Sample Input and Output of JavaRAP

Input:

Eastern Airlines executives notified union leaders that the carrier wishes to discuss selective wage reductions on Feb. 3.
 Union representatives who could be reached said they hadn't decided whether they would respond.
 By proposing a meeting date, Eastern moved one step closer toward reopening current high-cost contract agreements with its unions.

Annotated Input:

<COREF ID="11" MIN="representatives"><COREF ID="8" TYPE="IDENT" REF="9">Union<COREF> representatives who could be reached<COREF> said<COREF ID="10" TYPE="IDENT" REF="9">they<COREF> hadn't decided whether<COREF ID="12" TYPE="IDENT" REF="10">they<COREF> would respond.

Output:

- (2,0) Union representatives who could be reached <- (2,7) they, (2,7) they <- (2,12) they, (3,6) Eastern <- (3,18) its,
- Eastern Airlines executives notified union leaders that the carrier wishes to discuss selective wage reductions on Feb. 3.

Union representatives who could be reached< Union representatives who could be reached> had n't decided whether <Union representatives who could be reached> would respond.

By proposing a meeting date, Eastern moved one step closer toward reopening current high-cost contract agreements with <Eastern's> unions.

Sample Output of the Evaluator

...
Errors:
 [Complete text span > (anaphor) antecedent suggested by the resolver > minimal text span]*
 Yesterday's performance > (10,3 it)mr. callahan > performance
 *****PleoWrong 26,22 it --> the air line
 *****Missed: its
 *****Missed: they
 ...
 For "*/muc6/train/8703100059".
 26 11 1 8 2 4