



International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 7, July 2014

Automated Anaphora Resolution

Kalyani Kamune, Avinash Agrawal

Department Of Computer Science, Ramdeobaba college of engineering and management, Nagpur, India

ABSTRACT: Anaphora resolution has proven to be a very difficult problem of natural language processing, and it is useful in discourse analysis, language understanding and processing, information exaction, machine translation and many more. This paper represents an algorithm that instead of using a monolithic architecture for resolving anaphora, use the combination of constraint-based and preferences-based architectures, each uses a different source of knowledge, and proves effective on theoretical and computational basis. An algorithm identifies both inter-sentential and intra-sentential antecedents of “Third person pronoun anaphors”, “Pleonastic it”, and “Lexical noun phrase anaphora”. The algorithm use Charniak parser (parser05Aug16) as an associated tool, and it relays on the output generated by it. Saliency measures derived from parse tree, in order to find out accurate antecedents from the list of potential antecedents.

KEYWORDS: Anaphora, Anaphora resolution, antecedent, discourse, Pronominal Resolution

I. INTRODUCTION

Interpreting anaphoric expressions is one of the most fundamental aspects of language interpretation. The study of anaphora and anaphora has brought about many fundamental developments in theoretical linguistics and computational linguistics and has important practical applications in work on information extraction, summarization, and entity disambiguation. Anaphora resolution is a complicated problem in Natural Language Processing and has attracted the attention of many researchers. Anaphora describes the language phenomenon of referring to a previously mentioned entity (also called object or event); anaphora resolution is the process of finding that previous item. Consider the following clarifying example from a British World War II anti-raid leaflet:

“If an incendiary bomb drops next to you, don’t lose your head. Put it in any bucket and cover it with some sand.”

If this raised eyebrows - don’t worry - it is meant to. Indeed “it” could stand for (or refer to) either of the two objects mentioned before it, “bomb” and “head”. The authors meant the former, but the rules of language have a tendency to bias readers to picking the latter. But then “head’s” are not the usual things one puts in buckets and covers with sand. What anaphora resolution, when done correctly, enables us and systems to do, is to merge the previous information about an entity with the new information we encounter. So think of anaphora as the intricate balance between conciseness of communication and the ability of humans to understand each other. [1]

The remaining sections of this paper are organized as follows. In section 2, we represent the Literature Survey of Anaphora Resolution. In section 3, we represent in details how each step of “Automated Anaphora Resolution” algorithm is implemented. Section 4, explains the architecture of “Automated Anaphora Resolution”.

II. RELATED WORK

In the process of anaphora resolution, antecedents can be noun or verb phrases, any clauses, any sentence or even paragraphs/discourse segments as antecedents. Finding antecedents as a noun phrase is comparatively an easy task than that of finding rests as an antecedent. Generally, all noun phrases (NPs) preceding an anaphor are initially considered as potential candidates for antecedents. Search limit has to be predefined. An “ideal” anaphora resolution system has the search limit of 17 sentences away from the sentence in which anaphor is present.[4]



International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 7, July 2014

All the potential antecedents within the search limit preceding anaphora is find out, and various anaphora resolution factors are used to find the correct antecedent for the particular anaphora. Various factors used can be "eliminary" i.e. doesn't count certain noun phrases from the set of potential antecedents (such as gender , number , people constraints) or "preferential", assigning more preference to some potential antecedents and less to others (such as salience). [5]

System must defines the text segments in which the antecedent can be found, which is called as search limit or anaphoric accessibility space. This step is very important, for further processing. Keeping the search limit too narrow results in the exclusion of valid antecedents and keeping the search limit too broad results in the large candidate lists which is ultimately leads to erroneous results. Once the list of all possible candidates is found, several constraints can be applied in order to remove incompatible antecedents. It may be possible that after applying constraints, some anaphore still has more that one possible constraints, in such case we can get accurate constraints by using preferences.

Two types of architecture are present constraint-based architectures and preferences-based architectures, based on the factors (constraint-based or preferences-based) which we are using in the process of anaphora resolution. Instead of using a monolithic architecture for resolving anaphora, in the "Automated Anaphora Resolution" system we use the combination of constraint-based and preferences-based architectures; each uses a different source of knowledge, and proves effective on theoretical and computational basis. Hence the system mainly works on 3 steps given as: (1) defining an anaphoric accessibility space or Search Limit, (2) apply constraints, and then (3) apply preferences.

III. PROPOSED ALGORITHM

A. Design Considerations:

"Automated Anaphora Resolution" system resolves Anaphora by first reading input text, then it apply sentence splitter and adds tags like "<s> sentence </s>" around the splitter sentence. The output of sentence splitter is then given to the parser in order to generate "Tagged Text" and "Parse Tree". System then extracts the list of all noun phrases and a list of resolvable anaphors -- third person pronouns, reflexive Pronouns. Each anaphor is paired with all noun phrases within a small sentence window of 3 sentences.

For each anaphora and noun phrase obtained from the created list of anaphora and noun phrases, find agreement features like Number, People and Gender. In all three mentioned constraints, value of a particular constrain sets according to the predefined rules, otherwise, it remains "unknown".

- Agreement Filter

Pairs of noun and pronoun are then allowed to filter through 'Agreement filter'. The agreement features' compatibility of each and every pair of pronoun and a noun phrase is tested by an Agreement filter.

Required Information Obtained by Inspecting the Parse Tree Structure

All the information required by the system is not given by the parser; "Automated Anaphora Resolution" system recovers them by using structure information of the verb/noun phrases. [2]

- Pleonastic Pronouns Filter

System also identify pleonastic pronouns. In case of Pleonastic Anaphora, the pronoun doesn't have any referent. System uses the list of modal adjective and a cognitive verb in order to find pleonastic pronouns.

- Personal Pronoun Filter and Lexical Filter

"Automated Anaphora Resolution" system is used to find out the "Third person pronouns" in their nominative, accusative or possessive case and "Lexical anaphora". Information Obtained by Inspecting the Parse Tree Structure plays a vital role for identifying the "Third person pronoun anaphora" and "Lexical Anaphora".

- Salience weights

After applying all above mentioned filters, there is a possibility that for a particular anaphora more than one potential antecedent are present. Hence, in order to get final antecedent salience weight is used. [2]

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 7, July 2014

IV. SIMULATION RESULTS

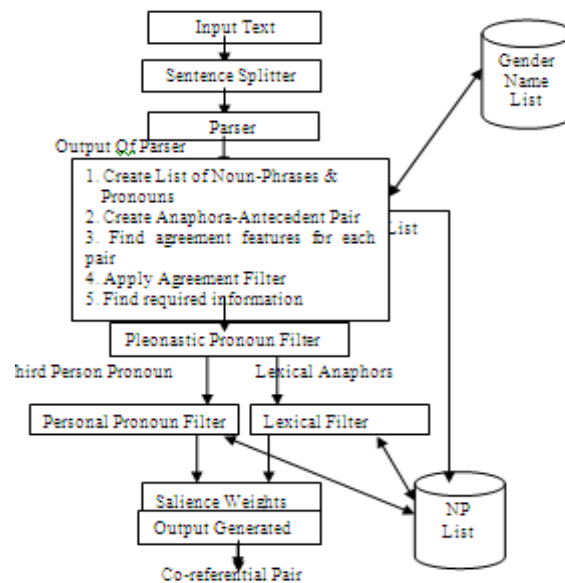


Fig. 1 Architecture of "Automated Anaphora Resolution".

An architecture for resolving anaphora, using combination of constraint-based and preferences-based architectures is given in the following figure. As per given in the figure, system will work on the syntactic structure generated by the parser, and parser accept the input generated by the sentence splitter. From the output generated by the parser, system will create two lists- list of pronoun and noun phrase. From the list of pronoun and noun phrases, all possible pairs of anaphora and antecedents are generated, and each pair is filtered through the agreement filter which works on the agreement features of each pronoun and noun phrase. All the information required by the system is not given by the generated parse tree, hence remaining required information is evaluated by the system for the further processing. Apply pleonastic pronoun filter which will take list of anaphora as an input. After applying pleonastic filter, personal pronoun filter and lexical filter is applied, this considers the list of anaphora and list of noun phrase generated by the system. There is a possibility that for a particular anaphora more than one potential antecedent is present. So, from the list of all potential antecedents, final antecedent is chosen with the help of salience weight.

V. CONCLUSION AND FUTURE WORK

An algorithm which uses the combination of constraint-based and preferences-based architectures, for resolving anaphors is represented as above. It is used to identify inter-sentential and intra-sentential antecedents of "Third person pronoun anaphors", "Pleonastic it", and "Lexical noun phrase anaphora". An algorithm at first defines an anaphoric accessibility space, then applying constraints, and finally applying preferences. Anaphoric accessibility space is selected carefully which is not too broad or too small. Several constraints are applied in order to remove incompatible antecedents for a particular anaphor. Finally, after removing incompatible candidates, if the remaining list contains more than one antecedent, salience weights are applied in order to choose a single antecedent. The algorithm applies to the output generated by Charniak parser (parser05Aug16) and relies on salience measures derived from parse tree. Various factors such as "eliminator" i.e. doesn't count certain noun phrases from the set of potential antecedents (such as gender, number, people constraints) and "preferential" i.e. assigning more preference to some potential antecedents and less to others (such as salience) is used in order to resolve the anaphora.



ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 7, July 2014

REFERENCES

1. Ruslan Mitkov, Anaphora Resolution: The State Of The Art, International Conference on Mathematical Linguistics, 2008
2. Shalom Lappin and Herbert J. Leass ,An Algorithm for Pronominal Anaphora Resolution, 1994
3. Chinatsu Aone and Scott William Bennett, Evaluating Automated and Manual Acquisition of Anaphora Resolution Strategies, International Workshop on Sharable Natural Language Resources (SNLR),2000
4. Mitkov and Ruslan, Anaphora resolution in Natural Language Processing and Machine Translation. Working paper. Saarbrücken: IAI, 1995a.
5. Mitkov, Ruslan, "Factors in anaphora resolution: they are not the only things that matter. A case study based on two different approaches" Proceedings of the ACL'97/EACL'97 workshop on Operational factors in practical, robust anaphora resolution., 14-21. Madrid, Spain, 1997b.