Journal of Computer Science 3 (4): 199-203, 2007
ISSN 1549-3636


# A Modular Architecture for Anaphora Resolution

Allaoua Refoufi
Computer Science Department, University Ferhat Abbas of Setif, Algeria

**Abstract:** Anaphora resolution attempts to determine the correct antecedent of an anaphor (the term pointing back). In what follows, we propose an algorithm for the resolution of anaphoric pronouns that relies on lexical and syntactic knowledge incorporated in a modular approach based on constraints and preferences. Our objective was to find the correct antecedent to the following subject pronouns (il, ils, elle, elles), object pronouns (l', le, la, les) and possessive pronouns (son, sa, ses, leur, leurs) in unrestricted texts. We also identify and eliminate pleonastic pronouns and discard candidates appearing in appositions. Moreover we use a focus mechanism to determine salient entities. The algorithm, implemented in Prolog, realizes a success rate of 68%, which was considered a good performance for unrestricted French texts.

**Keywords:** Anaphora, syntax, semantics, reference, focus

## INTRODUCTION

Anaphora describes the dependence of an expression on a previously mentioned one in a discourse segment. It is an important phenomenon required in almost every natural language application. Anaphora is used to explicitly exhibit relations between different linguistic units that designate the same entities, that is are co referential. The object that is being referred to is called the *antecedent*, the expression that refers to the antecedent is called the *referring expression* or *the anaphor*. The process of finding the proper antecedent for each anaphora in texts is termed *anaphora resolution*. In French, anaphoric expressions are signalled by two linguistic categories: pronominal anaphora and definite descriptions.

Example

« *Mon frère* m'appela hier, il voulait me voir" ("*My brother* called last night, he wanted to see me")

"*He*" is the anaphor; "*my brother*" is the antecedent. In general several antecedents are possible for the same anaphor. The algorithm identifies noun phrase antecedents of personal, demonstrative and reflexive pronouns in French. Pleonastic pronouns are also considered, since we have to discard them before proceeding further. An example of a pleonastic pronoun in French is "*il est important de bien manger*". The pronoun "il" does not refer to any entity. The strategy identifies both intrasentential (when the anaphor and the antecedent occur in the same sentence) and intersentential (when the anaphor and the antecedent do not occur in the same sentence) antecedents and is applied to the output of the syntactic analysis generated by a robust parser. The strategy combines different forms of knowledge and distinguishes between constraints and preferences. Whereas constraints are used as conditions that must not be violated, preferences are heuristic rules that sort the remaining candidates in an order, which is believed to be optimal.

Pleonastic pronouns are not considered anaphoric (since they don't have an antecedent), identifying such occurrences is important so that the anaphora resolution system will not try to look for their antecedents. When performing anaphora resolution, all noun phrases are typically treated as potential candidates for antecedents. The scope is usually limited to the current (when dealing with reflexive anaphora) and the two preceding sentences (for other types of anaphora) and all candidates within that scope are considered in turn. Appositives, also termed insertions in the French literature, are usually used to provide some additional information for a named entity. The additional information is separated from the name of the entity by a comma and is usually placed immediately after the entity name. For example: "*Caesar, the roman emperor, died in 44"*.

Appositional phrases are considered to supply additional information which is not of great interest in the text. The identification of appositives enables us to eliminate candidates which occur inside the apposition.

Anaphora resolution requires multiple sources of knowledge. Morphological analysis tells us how to extract the base forms out of inflected forms that occur in texts. This type of analysis is especially important for

**Corresponding Author:** Allaoua Refoufi, Computer science department, university Ferhat Abbas of Setif, Algeria

French where inflected forms of verbs proliferate. Syntax is concerned with the ways words combine to form phrases and phrases combine to form sentences. Syntax associates some structure to the utterances of the language; moreover it tells us the syntactic function of each word (verb, noun, pronoun, etc.).

The c-command constraint plays a crucial role in any anaphora resolution system because it provides an elegant way to discard noun phrases which syntactically cannot be potential candidates. Therefore it is part of the resolution process, first eliminate those entities which cannot be antecedents, then collect the ones who can be and proceed. For example in the sentence "Sarah likes her" it is obvious that the anaphor "her" does not refer to the antecedent "Sarah"; otherwise we would have written "Sarah likes herself". In French the distinction is stronger and would be "Sarah l'aime" and "Sarah s'aime" respectively. Equivalently for the sentence "Sarah admire la fille". "la fille" cannot be the referent to "Sarah". Formally speaking we say that "Sarah" and "la fille" are bounded, therefore they cannot co refer. This notion of disjoint reference is defined by the c-command constraint which states that a node A c-commands a node B in the parse tree if and only if: i)A does not dominate B, ii) B does not dominate A and iii) the first branching node dominating A also dominates B. C-command constraints are determined by the syntax of the text; that is the structure of the parse tree.

Although there exist several ways in which to define the c-command constraint, we have opted for a simple one[1]. One way to implement c-command constraints is to assign numbers to the nodes in the parse tree while parsing, each node c-commanded will have a number inferior to the number assigned to the node that dominates it.

**Previous work:** A pioneer work reported by Hobbs[2] uses a syntactic tree to search the input sentence for antecedents. The algorithm is a left to right breadth first search on the syntactic parse tree of the input sentence. Given the usual order of syntactic categories in the English language (the subject is generally followed by the verb) the algorithm expresses a preference for the noun phrases subjects.

Probably one the well known algorithm for pronoun resolution was proposed by Lappin and Leass[3]. The algorithm exploits salience factors and their associated weights such as sentence recency, subject emphasis, head noun emphasis) and so on to perform pronominal resolution. The salience value is simply the sum of the associated weights. Once salience values have been calculated for each referent, the algorithm can be applied to resolve the pronouns. The entity with the highest salience value is declared to be the most likely referent. If there are no pronouns to be resolved in a sentence, the next sentence is processed and the weights that contribute to an entity's salience are halved (to account for sentence recency). The weights used in the salience algorithm are ad hoc. Lappin and Leass's algorithm for pronominal anaphora resolution is capable of high accuracy, but requires in depth, full, syntactic parsing of text. The author's report 86% successfully identified antecedents in a corpus containing technical manuals.

Kennedy and Boguraev[4] describe a variant that does not require in-depth, full syntactic parsing of text. Instead, with minimal compromise in output quality, the modifications enable the resolution process to work from the output of a part of a speech tagger, enriched only with annotations of grammatical function of lexical items in the input text stream. Their method has been applied to personal pronouns, reflexives and possessives. The general idea is to construct co reference equivalence classes that have an associated value based on a set of ten factors. An attempt is then made to resolve every pronoun to one of the previous introduced discourse referents by taking into account the salience value of the class to which each possible antecedent belongs. The authors report 75.5 % success in resolution on a corpus containing texts of different genres.

Mitkov's algorithm[5] is another knowledge poor approach to pronominal resolution, which means that it uses only the output of a part of speech tagger with minimal syntactic information. The algorithm does not employ syntactic information but relies on a set of indicators (rules) such as definiteness, heading, collocation, referential distance, term preference, etc. The indicators, boosting and impeding ones, assign salience values to the antecedents. The boosting indicators assign a positive score to a noun phrase, reflecting a positive likelihood that it is the antecedent of the current pronoun. In contrast, the impeding ones apply a negative score to a noun phrase, reflecting a lack of confidence that it is the antecedent of the current pronoun. A score is calculated based on these indicators and the discourse referent with the highest aggregate value is selected as antecedent. The author reports success rate of 89.7% on a corpus of technical manuals. The main drawback of the systems that use full parsing is that their performance appears to plateau at around 60-65% on unrestricted text[6].

**The resolution method:** Our system is composed of two main tasks: the first one, the recognition phase,

performs parsing, structure building, recognition of pleonastic pronouns and identification of focusing expressions and identification and elimination of appositives.

The second part, the anaphora resolution procedure, applies the constraints and the preferences. The constraints used in the algorithm are the following: morphological agreement (gender, number and person) and c-commands restrictions. A preference is a characteristic that is not always satisfied by the solution of an anaphor. The aim of the preference heuristics is to obtain a ranked list of candidates. The preferences used in our system are: syntactic parallelism, antecedent not included in a prepositional phrase, focused expressions and recency.

Non anaphoric expressions are signalled by expressions of the form « il est {possible, évident, admis, normal, pertinent, logique, courant, etc.} que … » and indicate that the pronoun « *il* » is not anaphoric. The identification of non anaphoric pronouns is based a simple pattern matching procedure. The number of template matching used to identify such constructions can be updated or augmented.

Focussing is defined as the process which chooses a theme, or center of attention, in a discourse and moves it as the speaker's discourse proceeds. The focus provides a valuable source for identifying pronominal anaphora. Focusing expressions are of the form "c'est NP qui", "il y a NP", where the noun phrase NP is the focus[7,8]. The focusing mechanism adopted is rather simple: if the focus is not explicitly stated in a focusing expression, it is set to the subject of the previous sentence. After each anaphor resolution the focus is set to the antecedent selected. Initially the focus is set to the subject of the current sentence.

**The constraints:** Constraints are rules which participate in the purging of the candidates appearing in the structures built during the parsing process. Incorporated constraints are: syntactic constraints and consistency conditions

Syntactic constraints are based on c-command relations; they eliminate noun phrases that cannot be antecedents. Consistency conditions are agreement on morphological grounds (gender, number and person). For reflexive pronouns, the potential candidates appear in the same sentence as the anaphor.

**Preferences:** Preferences, as opposed to constraints, can be violated by the antecedent candidates; they are used to rank the candidates. However those that verify the preferences are retained. The order in which they appear reflects their weight. The preferences incorporated are:

1. Candidates in the focus register (preference(1))
2. Syntactic parallelism (preference(2))
3. Antecedent not occurring in a prepositional phrase (preference(3))
4. Recency (preference(4))

Syntactic parallelism states that we prefer the antecedent that shares the same syntactic function as the anaphor. For example:

« *L'enfant* reconnut *le roi* ; pourtant il ne l'avait jamais rencontré auparavant ».

« *The child* recognized *the king*; although he has never met him before".

The antecedent of the anaphor "il" is "l'enfant".

An expression, mainly a noun phrase, included in a prepositional phrase is unlikely to be referred to because it only brings additional information. For example in the sentence : « *La voiture* de la voisine nous bloque le chemin, il faut la déplacer » ; the anaphor « *la* » refers to «la voiture» and certainly not to «la voisine ».

The recency preference favours the candidate which lies nearest to the anaphor; that is the one evoked recently.

**Algorithm design**

**Recognition phase**

1. Parse sentence i to get the parse tree i (initially i=1)
2. Recognition of non anaphoric pronouns
3. Identification and elimination of appositives
4. Identification of focusing expressions, updating the focus register
5. Structures building, update the potential candidates list

**Resolution phase:** Let L(i), L(i-1) and L(i-2) be the lists of the potential antecedents from sentences i, i-1 and i-2; sentence i being the current one.

Let the list of anaphors be $a_1$, $a_2$, …, $a_k$ for the current sentence i ( in any case k does nod exceed 3).

Let $E_1$ be the set {L(i), L(i-1), L(i-2)} (concatenation of the 3 lists)

For each anaphor α

Check the couple {α, m} for syntactic conditions; where m is in L(i)

Update L(i)

For each p in $E_1$ check {α, p) for morphological agreement to get the set $E_2$

For k = 2 to 4 do

begin

If $| E_k | = 1$ stop; output $(\alpha, E_k)$

If $| E_k | > 1$ apply preference(k-1) to get $E_{k+1}$

end
If $|L_5| = 1$ stop; output$\{\alpha, L_5\}$
If $|L_5| > 1$ set the solution to the first element of $L_5$.
Updating the potential candidates lists
At each step, the list of potential candidates E contains noun phrases from sentence i, (i-1) and (i-2).

1. Remove from E noun phrases from sentence (i-3)
2. Insert in E noun phrases from the current sentence i

The overall strategy of the algorithm is as follows: for each anaphor encountered construct a list L of the potential candidates, checking for syntactic and agreement constraints. If the list L contains more that one element, apply in turn the preferences. As soon as a preference is satisfied the algorithm stops.

## DISCUSSION

For the time being the parser has a wide linguistic coverage of French syntax and it uses a dictionary of about 2500 words. Proper names and named entities are only accepted if they belong to the domain lexicon, which can be updated easily.

Our objective is to process the following subject pronouns (il, ils, elle, elles), the object pronouns (l', le, la, les), possessive pronouns (son, sa, ses, leur, leurs). These pronouns are frequent in literary texts.

When using the pronoun « l' » we face more difficulties because the gender feature is missing. We also encountered the case where the gender feature of the pronoun "*elle*" is of little help in the determination of the correct antecedent, as in the example

"*Le docteur Bouzidi* est un spécialiste en chirurgie. Elle travaille jour et nuit*"*.

"Doctor Bouzidi is a specialist in surgery. She works night and day".

The pronoun « lui » can be either a masculine or feminine indirect object, as in the example

« Elle lui donne la clé *»*.

Our algorithm does not deal with anaphors that refer to verb phrases or sentences, as in :

« Sur un deux roues, *on est très fragile*. Le problème c'est de l'oublier ».

"On two wheels we are vulnerable. The problem is to forget it".

The algorithm realises a success rateTPPT of 68 %, the corpora used is extracted from literary textbooks where the phenomena of anaphora is very dense, as opposed to scientific texts. Texts used consist of 3 to 5 chapters, each chapter contains about 5 sentences, each sentence contains 5 to 20 words. The evaluation has been carried out so far on 250 texts of reasonable size.

The results show that the resolution of pronouns such as *il(s), elle(s)* is relatively successful (success rate of 93 %).

Our system is capable of identifying two types of appositions: those located between parentheses or brackets and those located between commas. The implementation for the latter apposition is a bit complex, as we have to make sure that the final comma does exist before deciding whether we are in the presence of an apposition. When recognized, appositives are simply jumped over.

The insertion constraint tends to add more complexity in the implementation, which to the best of our knowledge does not carry real improvement to the algorithm.

Attachment ambiguities problems of prepositional and adverbial clauses during the parsing process are resolved using the minimal attachment principle which stipulates that we favour the parse tree with the minimal number of nodes.

## CONCLUSION

The main idea of our work consists of the establishment of a link between nominal phrases that share similar context with constituents in the input text. The method relies heavily on a morpho syntactic robust parser, where the main knowledge is gathered. The overall strategy of the algorithm is as follows: for each anaphor encountered construct a list L of the potential candidates, checking for syntactic and agreement constraints. If the list L contains more that one element, apply in turn the preferences. As soon as a preference is satisfied the algorithm stops. The application of a set of constraints followed by a set of preferences provides an elegant modular, easy to update anaphora resolution algorithm. We believe that only a full syntactic parsing can provide the required knowledge to the anaphora resolution module. The performance of the algorithm implemented cannot be compared to the reviewed ones because the language is French in our case. Furthermore the rate of success depends on the data used. In our case we have used literary stories where the density of anaphoric expression is rather high. In comparison with previous work, the success rate realised (68 %) is very satisfactory and motivates us to explore new ways in which to improve our work.

## REFERENCES

1. Mitkov, R., 2002. Anaphora Resolution. Longman, London.

2. Hobbs, J.R., 1978. Resolving pronoun references. Lingua, 44: 339-352.
3. Lappin, S. and H.J. Leass, 1994. An algorithm for pronominal anaphoric resolution. Computational Linguistics, 20: 535-561.
4. Kennedy, C. and B. Boguarev, 1996. 1996. Anaphora for everyone: Pronominal anaphora resolution without a parser. Proc. 16th Intl. Conf. on Computational Linguistics (COLING'96), pp: 113-118. Copenhagen, Denmark.
5. Mitkov, R., 1998. Robust pronoun resolution with limited knowledge. Proc. 18th Intl. Conf. on Computational Linguistics, Montreal, Canada.
6. Preiss, J., 2002. Choosing a parser for anaphora resolution. Proc. 4th Discourse Anaphora and Anaphor Resolution Colloquium (DAARC, 2002), Lisbon, Portugal, pp: 175-180.
7. Depain-Delmotte, F., 1999. la Sélection de l'Antécédent du Pronom dans les Systèmes de Traitement Automatique des Langues Naturelles, Proceedings of Vextal'99 Venezia San Servolo.
8. Refoufi, A., 2005. A multiple knowledge source algorithm for anaphora resolution. Invited paper, Natural Language Group, Computer Science Department, University of Sheffield, England, Nov. 24. http\ shef. ac. Uk \talks