

A Comparative Analysis of Centering-Based Algorithms for Pronoun Resolution in Portuguese

Fernando José Vieira da Silva¹, Ariadne Maria Brito Rizzoni Carvalho¹,
and Norton Trevisan Roman²

¹ Institute of Computing, State University of Campinas, Campinas, Brazil
ra085324@students.ic.unicamp.br, ariadne@ic.unicamp.br

² School of Arts, Sciences and Humanities – University of São Paulo,
São Paulo, Brazil
norton@usp.br

Abstract. Pronominal anaphora resolution consists in finding a referent for a given pronoun. Although being essential for many natural language processing systems, such as automatic translators, text generators and summarizers, there is a myriad of issues regarding this task, particularly when there is more than one possible referent for a given pronoun. Over the years, several approaches have been proposed to deal with this challenge, usually taking only syntactic information into account. On the other hand, methods based on Centering Theory rely on concepts such as coherence, for instance, to do the job. In this work, we describe our implementation and evaluation of existing centering-based algorithms for pronominal resolution in Portuguese. As a result, we indicate both the strong and weak points of each of the tested algorithms, thereby helping other researchers to make a more informed decision about which method to use.

1 Introduction

Anaphora is a linguistic phenomenon that happens when there is an abbreviated reference to another element in the text. This abbreviation is called an *anaphor*, while the element to which it refers is called a *referent*. Although, in this work we focus only on pronominal anaphora, i.e., when the anaphor is a pronoun, the challenge of finding a referent for a given anaphor still remains a great one, raising considerably in the presence of more than one possible referent, as in the sentence: “O **cachorro** de João fugiu esta manhã. **Ele** latiu e assustou o carteiro.” (“João’s **dog** fled this morning. **It/He**¹ barked and scared the post-man.”). In this sentence, both “cachorro”(dog) and “João” are candidates for the anaphor “Ele”, since both agree in gender and number with the pronoun, being this kind of ambiguity solved at the semantic level only.

On the search for solutions to automatic pronominal anaphora resolution, several strategies have been explored. The approaches studied in this work are

¹ In Portuguese, both “it” and “he” are translated to “ele”.

based on Centering Theory [8], which, among other things, deals with the notion of local discourse coherence, i.e., coherence between utterances [9], as determined by the way the information is presented in the text. Therefore, Centering Theory provides some insights to the pronoun resolution task, by always looking for a referent that would keep the discourse's coherence, there still existing, when in the absence of such a referent, the intuition that such discourse segment is not coherent.

In this work, we describe the results obtained on implementing and analyzing such algorithms by running them over a small corpus of texts adapted from the papers in which the algorithms are presented, along with other papers that deal with Centering Theory (*e.g.* [5,3]). By doing so, we intend to provide other researchers with information that may help them in the choice for the most suitable algorithm, according to the properties of the analyzed corpus. We also discuss about each method's main features, describing not only their precision, but also their computational complexity. Based on our results, we also make some suggestions for modifications in these algorithms.

The remaining of this paper is organized as follows: the next section presents the Centering Theory and its concepts for pronominal anaphora resolution; Section 3 shows a conceptual algorithm, purely based on the ideas introduced by the theory, as described in [8] and [3], designed to serve as a basis for comparison to the other algorithms. The implementation of the BFP, S-List and LRC algorithms are described in Sections 4, 5 and 6, respectively. In Section 7, we discuss some characteristics and behaviour of these algorithms and finally, Section 8 shows our conclusions and final comments.

2 Centering Theory

Centering Theory is a system of rules and constraints that govern the relations between the text goals and the author's linguistic choices to express the flow of ideas [1]. Based on Grosz and Sidner's Theory [7], Centering Theory introduces the notion of local coherence, i.e., coherence between utterances in the same discourse segment. Discourse coherence is then measured by the number of times the writer changed the subject in the text, depending on the sequence the information is presented.

According to Centering Theory, the participants' focus of attention can be captured by some "centers", which exist in every utterance of the discourse. Let U_n be the n -th discourse utterance, and U_{n-1} its precedent; each utterance U_n has the following associated elements [1,7,8]:

- **$Cf(U_n)$ (Forward Looking Center)**: It is composed of every entity which may be the focus on the next utterance, consisting of all pronouns and noun phrases found in the current utterance. The elements in the $Cf(U_n)$ are ordered according to their syntactic function in the text, with the following preference: subject > object > others;
- **$Cp(U_n)$ (Preferred Center)**: It is the most probable element to be the center in the next utterance. The Preferred Center is the best ranked element in $Cf(U_n)$;

- **$Cb(U_n)$ (Backward Looking Center)**: It corresponds to the best ranked element from the previous utterance ($Cf(U_{n-1})$) which is realized as a pronoun in the current utterance.

In order to measure the text coherence, as indicated by the change of focus through utterances, Centering Theory specifies all center transitions according to the change or maintenance of entities that compose the $Cf(U_n)$, $Cp(U_n)$ and $Cb(U_n)$ of each utterance. The transitions considered in this work, including the Shifting-1 transition introduced in [3], are:

1. **Center Continuation**: This transition indicates that the information about the same entity is presented consecutively, without any other entity in the discourse; i.e., the focus of attention is maintained. Formally, it occurs when: $Cb(U_n) = Cb(U_{n-1})$ and $Cb(U_n) = Cp(U_n)$;
2. **Center Retaining**: Although there exists a reference to another entity in the discourse, its focus is kept the same, thereby continuing the same theme. It occurs when $Cb(U_n) \neq Cp(U_n)$ and $Cb(U_n) = Cb(U_{n-1})$;
3. **Shifting-1**: The discourse focus is slightly shifted, changing the text theme. It is expected, however, that this new theme will be maintained in the next utterance. It happens when $Cb(U_n) \neq Cb(U_{n-1})$ and $Cb(U_n) = Cp(U_n)$;
4. **Center Shifting**: It indicates a change in the discourse focus. It occurs when $Cb(U_n) \neq Cb(U_{n-1})$ and $Cb(U_n) \neq Cp(U_n)$.

Based on the identified center transition in a specific discourse segment, one can determine its coherence, by considering a segment with Center Continuation more coherent than another with Center Retaining which, in turn, is more coherent than one with Shifting-1. Finally, segments with Center Shifting are considered the less coherent among all. The intuition behind this rule – which in fact has some experimental support (*e. g.* [3]) – is that a text that shows information about the same entity in a sequential way is more coherent than another that mentions several entities in a interleaved way.

Finally, along with defining a measure for discourse coherence, Centering Theory also presents some rules regarding pronoun realization, to wit, if some element of $Cf(U_{n-1})$ is realized by a pronoun in $Cf(U_n)$, then this element must be the $Cb(U_n)$. When the utterance has several pronouns, one of them must be the $Cb(U_n)$. If, however, there is only one, then it must be the $Cb(U_n)$.

3 Conceptual Algorithm

Since Centering Theory was not specifically designed for pronominal resolution, it does not deal with some related problems, such as finding referents for intra-sentential anaphora, for example. Due to this fact, algorithms like BFP, S-List and LRC need some customizations in order to handle these issues, or even to become more accurate. In order to evaluate these customizations, we have compared these three algorithms with a fourth one, named Conceptual Algorithm, built exclusively on the concepts introduced by Centering Theory, and which should be used as a benchmark.

-
1. Eu não vejo João há alguns dias.
 (I haven't seen João for a few days).
 $Cf = \{Eu, João, dias\}$
 $Cb = \{\}$
 $Cp = \{Eu\}$
2. Carlos acha que ele está estudando para suas provas.
 (Carlos thinks he is studying to take tests).
 $Cf = \{Carlos, Ele = João, provas\}$
 $Cb = \{João\}$
 $Cp = \{Carlos\}$
 Transition = CENTER SHIFTING
3. Eu acho que ele foi para o interior com Linda.
 (I think he went to the countryside with Linda).
- (1st Set : ele=Carlos)
 $Cf = \{Eu, ele, interior, Linda\}$
 $Cb = Carlos$
 Transition = CENTER SHIFTING
- (2nd Set - THE CHOOSEN ONE! : ele=ele=João)
 $Cf = \{Eu, ele, interior, Linda\}$
 $Cb = ele$
 Transition = CENTER RETAINING
-

Fig. 1. Discourse sample and sets built by Conceptual Algorithm

In a nutshell, the algorithm builds a universal Cf set for each utterance. This set contains every entity mentioned in the utterance, grammatically ordered according to the following preference [3]: subject > object > object 2 > others > adjunct. Then, for each pronoun in the utterance, the algorithm builds all possible $\langle Cf, Cb \rangle$ sets, where Cb is some entity from $Cf(U_{n-1})$ which agrees in gender and number with the pronoun under consideration. In this implementation, a different set is built for each entity in $Cf(U_{n-1})$ that agrees with this pronoun.

The algorithm then chooses one of the $\langle Cf, Cb \rangle$ sets for each pronoun it finds in the current utterance, by analyzing the center transitions for each set and choosing the most coherent one. If, however, there is more than one set with the same center transition, being this transition the most coherent, the algorithm chooses the one whose Cb is better ranked in $Cf(U_{n-1})$. This referent is then taken for the Cb of the chosen set. As such, the algorithm does not try to obtain a single $\langle Cf, Cb \rangle$ set for each utterance, but instead, it considers one different set for each pronoun in that utterance, being the corresponding Cb its referent.

When the algorithm chooses a set for some pronoun in the utterance, it seeks the referent for that specific pronoun only. Thus, even if there are other pronouns

in the utterance, they are still mentioned in the *Cf* of the set but with no indication of their referents². Therefore, the *Cb* of this set which, in turn, is the referent for the pronoun, is the only entity realized by a pronoun in the *Cf* of the set. Naturally, the *Cb* is the best ranked entity from $Cf(U_{n-1})$ that is realized in the *Cf*, which is in line with Centering Theory's rules.

Figure 1 illustrates the process of finding referents for pronouns in an utterance. In this figure, the third utterance exemplifies the variation of sets for the pronoun "ele" (he) and the algorithm's choice according to the center transition. In the example, two sets are created for the pronoun. The second one is chosen because it presents a Center Retaining transition between utterances 2 and 3, which is more coherent than the Center Shifting that would happen if the first set were chosen. So the algorithm concludes that the pronoun "ele" refers to "João". As for the algorithm's computational complexity, the task of finding referents for all pronouns of a given utterance is dominated by the step in which the algorithm chooses a set for every pronoun in that utterance, taking $O(n^3)$ time, where n is the number of entities realized in that utterance.

4 The BFP Algorithm – Brennan, Friedman e Pollard

The algorithm introduced by Brennan, Friedman and Pollard [3] was the first and possibly the best well-known algorithm based on concepts from Centering Theory [6], comprising the following steps:

1. Creation of all possible $\langle Cf, Cb \rangle$ sets for the utterance;
2. It filters the sets created in step 1;
3. It chooses a set with the highest coherence, as determined by Centering Theory.

In the first step, all entities in U_n are grammatically ordered, according to the following preference: subject > object > object 2 > others > adjunct. Then all possible *Cf* elements are created as follows: for each pronoun in the utterance, an expanded set is generated, composed by elements in the format $\langle en, ref \rangle$, where en is the pronoun under consideration and ref is a candidate for referent. This expanded set contains one element for each entity in $Cf(U_{n-1})$ that agrees in gender and number with the pronoun. Next, each possible *Cf* is generated, containing one element from each different expanded set and each noun phrase in the utterance. Finally, each possible *Cf* is combined with each *Cb* candidate, where every entity in $Cf(U_{n-1})$ and the empty element are possible *Cbs*. This combination results in a list of $O(nm^2)$ possible $\langle Cf, Cb \rangle$ sets, where n is the number of entities realized in the current utterance, and m is the number of entities realized in the previous utterance. The algorithm then filters and discards some sets from that list, according to the rules described below:

1. ***Cb* not realized:** Every set whose *Cb* is not realized by some entity in that set's *Cf* is discarded;

² The referents for these pronouns are found separately by the algorithm.

2. ***Cb* is not the best ranked element in *Cf*(U_{n-1}):** If the *Cb* element is realized by some entity in the *Cf*, but there is another entity which is better ranked in *Cf*(U_{n-1}) being also realized by a pronoun, then this set is discarded;
3. **Filter by contra-indexes:** If there are two contra-indexed³ pronouns in the *Cf*, referring to the same entity, or if there is a pronoun that refers to an entity with which it is contra-indexed, then this set is discarded.

After these filters are applied, the algorithm chooses a $\langle Cf, Cb \rangle$ set for the utterance, by taking the center transition for each possible remaining set and choosing the most coherent one. The algorithm's computational complexity is dominated by the filtering step, which takes the time $O(n^5)$.

5 The S-List Algorithm

This algorithm is characterized by the use of only one data structure to store the entities of the discourse – the S-List, which is composed by every entity realized in a specific utterance, ordered as described below:

1. **Information Status** – The Information Status of an entity indicates if it was recently introduced into the context or if it has been mentioned earlier. According to this criterion, entities are classified as OLD, MED or NEW, where the preference order for entities is $OLD > MED > NEW$. The classes that compose these groups are the following [12,13]:
 - **OLD:** Composed by entities classified as EVOKED or UNUSED (whose meaning will be made clearer later on);
 - **MED:** Formed by entities classified as INFERRABLE, CONTAINING INFERRABLE and ANCHORED BRAND NEW. It is assumed that CONTAINING INFERRABLE and ANCHORED BRAND NEW are equivalent in practice;
 - **NEW:** Only the BRAND NEW class belongs to this group;
2. **Sentence position in the discourse – where the entity is realized:** Entities closer to the end of the discourse are better ranked than the others;
3. **Position within the utterance:** Entities closer to the beginning of the utterance are better ranked than the others.

The algorithm then adds each entity to the S-List, respecting the ordering rules mentioned above. After processing an utterance, it discards the unrealized entities. The search for a referent then becomes a simple lookup through the S-List. The entities's classification according to their Information Status is as follows [16]⁴:

³ In this implementation, according to Binding Constraints defined by Chomsky [10], two pronouns or noun phrases are considered contra-indexed if both cannot refer to the same entity.

⁴ In [16], however, neither INFERRABLE or CONTAINING INFERRABLE are considered.

- **EVOKED:** If the entity is realized by a pronoun;
- **BRAND NEW:** If the entity is preceded by an indefinite article;
- **UNUSED:** If the entity is not preceded by a determiner⁵. Only proper names fall into this category;
- **ANCHORED BRAND NEW:** If none of the conditions above hold;

The S-List algorithm, implemented as described in [16], has $O(n^2)$ computational complexity for finding referents for all pronouns in a given utterance.

6 LRC Algorithm – *Left-Right Centering*

The LRC Algorithm, proposed in [15], appends every entity in the utterance to a partial *Cf*. When it finds a pronoun, the algorithm starts looking intra-sententially for a referent in this partial *Cf* from left to right. The first found entity that agrees in gender and number with the pronoun and, in addition, respects the Binding Constraints defined by Chomsky [10], is taken as referent. At the end, the pronoun is inserted in the partial *Cf*. If no suitable referent is found, then the algorithm looks for a referent in $Cf(U_{n-1})$, starting with the best ranked entity, also reading it from left to right. When the process is finished, the $Cf(U_n)$ is built by simply ordering the partial *Cf* grammatically, as in the BFP algorithm. This algorithm's computational complexity for finding the referent for every pronoun in a given utterance is $O(n^2)$.

7 Discussion

Since it is based only on Centering Theory's rules, the Conceptual Algorithm has no other strategy for choosing $\langle Cf, Cb \rangle$ sets than measuring center transition coherence. This fact leaves room for the possibility that, for an utterance having two pronouns that agree in gender and number with each other, the algorithm may choose the same referent for both, leading to incoherent interpretations in some cases. Along the same lines, there is also the possibility that the algorithm may assign the same referent to two contra-indexed pronouns [10]. The BFP algorithm, on the other hand, deals with contra-indexed pronouns through its filtering process.

Nevertheless, a common criticism to the BFP Algorithm concerns its computational complexity [15]. Overall, the responsibility for the exponential complexity can be mainly attributed to the $\langle Cf, Cb \rangle$ set generation, since this step creates $O(nm^2)$ sets, which must be considered by all forthcoming steps, either at the filtering or final set identification stage. Another drawback present in both BFP and Conceptual algorithms is their incapability to solve intra-sentential anaphora, while S-List and LRC may solve them correctly.

Concerning intra-sentential anaphora, unlike the S-List Algorithm, LRC always prefers intra-sentential referents to inter-sentential ones, which might be

⁵ Determiners are definite and indefinite articles and pronouns tagged as determiners by the *PALAVRAS* [2] parser.

harmful to its success rate, when analyzing a text containing mostly inter-sentential anaphora. On the other hand, its ordering strategy allows the S-List Algorithm to choose an inter-sentential referent instead of a possible intra-sentential one, depending on its Information Status. Such a mechanism may improve its accuracy over an heterogeneous corpus, with a balanced number of intra- and inter-sentential anaphora. The ability for solving intra-sentential anaphora, however, does not necessarily means an advantage for the algorithms S-List and LRC, since in texts where inter-sentential anaphora are majority, BFP and Conceptual algorithms may have advantage.

In this work, we also run these algorithms to find referents for personal pronouns over the Summ-it corpus [4], which is composed of articles extracted from the science section of a well known Brazilian newspaper, containing 118 personal pronouns in total, with 7% of intra-sentential anaphora. The articles are segmented in text spans used to define RST⁶ relations [14] and we consider these spans as utterances. In our implementation, only nouns and personal pronouns are available to become part of Cf, Cb or S-List, where each noun is considered as a realization of a different entity and each personal pronoun is considered as a realization of a previous mentioned entity. For this evaluation however, we haven't implemented an algorithm for checking Binding Constraints, so none of the algorithms check them. The results are shown in Table 1. In this table, "Intra Result" and "Inter Result" columns give, respectively, the percentage of success in finding referents for intra-sentential and inter-sentential anaphora⁷, while the column "Overall Result" presents the percentage of success in finding referents for any kind of anaphor. The "Not Found" column indicates the percentage of anaphora to which the algorithm could not find any referent.

Table 1. Results obtained from running Conceptual, BFP, S-List and LRC algorithms over the Summ-it corpus

Algorithm	Complexity	Intra Result	Inter Result	Not Found	Overall Result
Conceptual	$O(n^3)$	0%	37%	31%	34%
BFP	$O(n^5)$	0%	37%	31%	34%
S-List	$O(n^2)$	78%	34%	25%	37%
LRC	$O(n^2)$	78%	34%	25%	42%

As expected, both Conceptual Algorithm and BFP did not solve any intra-sentential anaphora. On the other hand, S-List and LRC have shown a much better success rate for solving intra-sentential anaphora than for inter-sentential ones. However, inter-sentential results for the two first algorithms are better than for the last two. Still, overall results for S-List and LRC are better than for the algorithms that do not solve intra-sentential anaphora, even for this corpus where inter-sentential anaphora are majority. In this corpus, the highest success rate is reached by the S-List algorithm, whose advantage in comparison

⁶ Rhetorical Structure Theory.

⁷ The success rate is relative to the number of intra- and inter-sentential anaphora.

to LRC may be explained by the high frequency of inter-sentential anaphora in the corpus, since LRC always prefers intra-sentential referents. As we did not check for binding constraints, the equality between Conceptual and BFP results is comprehensible.

Finally, none of the algorithms is supposed to find referents that are not realized in the previous utterance, which, in turn, is a limitation of the Centering Theory, when used for anaphora resolution. We believe that it may be the main reason for the cases where the algorithms were not able to find a referent.

8 Conclusion

In this work we presented four algorithms based on Centering Theory for automatic pronominal anaphora resolution, describing their implementations. We also analyzed their main features, along with some possible situations where each of them would succeed or not. Results show that, overall, the best suited algorithm for a corpus with a balanced number of intra-sentential and inter-sentential anaphora is S-List, due to its ability to solve intra-sentential anaphora without discarding inter-sentential referent candidates. Even in our experiments, where we used a corpus with a much higher number of inter-sentential anaphora, S-List still kept the best success rate.

Another interesting result from our experiment is the high number of anaphora that the algorithms were not able to resolve. A deeper study of these cases might lead to thorough modifications on algorithms based on Centering Theory for anaphora resolution, by having algorithms that consider, as referent candidates, entities realized in utterances other than the one immediately before the anaphor.

Other avenues for future work include the evaluation of some variations of the originals S-List and LRC algorithms. For example, when two entities in an utterance have the same Information Status, S-List could take the entity which is closer to the pronoun, instead of preferring the one closer to the beginning of the utterance. As for LRC, we might try to implement the intra-sentential search in the opposite way, starting from right to left. The reason to believe these approaches could result in improvements for the algorithms is the fact that Hobb's Algorithm [11] presents such a preference, and experiments made in [15] found a better success rate in a corpus with more than 71% of intra-sentential anaphora. Finally, the S-List algorithm makes no distinction between two entities with different Information Status if they belong to the same group. Take, for example, the UNUSED and EVOKED classes. Currently, entities belonging to these classes have the same weight in S-List, being part of the OLD group. However, it could be the case that it is more probable that an EVOKED entity be realized again in the text than an UNUSED entity be realized for the first time, in which case it might be worth distinguishing one from the other. Practical experiments on distinguishing entities with different Information Status in the same group would be interesting.

References

1. Aires, A., Coelho, J., Collovini, S., Quaresma, P., Vieira, R.: Avaliação de centering em resolução pronominal da língua portuguesa. In: 5th International Workshop on Linguistically Interpreted Corpora of the Iberamia 2004, Puebla, Mexico, pp. 1–8 (November 2004)
2. Bick, E.: The Parsing System “Palavras”. In: Automatic Grammatical Analysis of Portuguese in a Constraint Grammar Framework. University of Aarhus, Aarhus (2000)
3. Brennan, S.E., Friedman, M.W., Pollard, C.: A centering approach to pronouns. In: ACL 1987, pp. 155–162. ACL, Stanford (1987)
4. Carbonel, T.I., Collovini, S.S., Coelho, J.C., Fuchs, J.T., Rino, L.H.M., Vieira, R.: Summ-it: Um corpus anotado com informações discursivas visando à sumarização automática. In: Proceedings of XXVII Congresso da SBC: V Workshop em Tecnologia da Informação e da Linguagem Humana? TIL, Rio de Janeiro, Brazil, pp. 1605–1614 (2007)
5. Cristea, D.: Discourse theories and technologies. In: Tutorial at ICON 2004, Hyderabad, India (2004)
6. Grosz, B.J., Joshi, A.K., Weinstein, S.: Towards a computational theory of discourse interpretation (1986)
7. Grosz, B.J., Sidner, C.: Attention, intentions, and the structure of discourse. *Computational Linguistics* 12(3) (1986)
8. Grosz, B.J., Weinstein, S., Joshi, A.K.: Centering: A framework for modeling the local coherence of discourse. *Computational Linguistics* 21, 203–225 (1995)
9. Poesio, M., Eugenio, B.D., Stevenson, R., Hitzeman, J.: Centering: A parametric theory and its instantiations. *Computational Linguistics* 30, 309–363 (2004)
10. Santorini, B., Kroch, A.: The syntax of natural language: An online introduction using the Trees program (2000), <http://www.ling.upenn.edu/~beatrice/syntax-textbook>
11. Santos, D.N.A., Carvalho, A.M.B.R.: Hobbs’ algorithm for pronoun resolution in portuguese. In: Gelbukh, A., Kuri Morales, Á.F. (eds.) MICAI 2007. LNCS (LNAI), vol. 4827, pp. 966–974. Springer, Heidelberg (2007)
12. Strube, M.: Never look back: an alternative to centering. In: Proceedings of the 17th International Conference on Computational Linguistics, pp. 1251–1257. Association for Computational Linguistics, Morristown (1998)
13. Strube, M., Hahn, U.: Functional centering – grounding referential coherence in information structure. *Computational Linguistics* 25, 309–344 (1999)
14. Taboada, M., Mann, W.C.: Introduction to rst (rhetorical structure theory), <http://www.sfu.ca/rst/01intro/intro.html>
15. Tetreault, J.R.: Analysis of syntax-based pronoun resolution methods. In: ACL 1999, pp. 602–605. College Park, MD (1999)
16. Tetreault, J.R.: A corpus-based evaluation of centering and pronoun resolution. *Computational Linguistics* 27, 507–520 (2001)