An Algorithm to Calculate Semantic Distance among Indirect Question Referents

Lokesh Kumar Sharma¹ and Namita Mittal²

Department of Computer Science and Engineering
Malaviya National Institute of Technology Jaipur, India
¹2013rcp9007@mnit.ac.in, ²namittal.cse@mnit.ac.in

Abstract

In question answering, anaphora resolution is an important part while extracting an answer in unstructured text. Indirect anaphora is defined as the referent-referee relation between two somehow related entities (e.g. car and garage). These referents of indirect referring expressions are unstated and also not mentioned previously in the text. Distance between these co-refers can be calculated which can be used at the time of semantic tagging. Which identifies the most probable hidden word associated with the entity. In the proposed algorithm, semantic distance of two referents is calculated using the concept of tree traverse algorithm. The calculated distance between two referents is used for suggesting the accurate hidden word to resolve indirect anaphora in answer source. Algorithm is discussed for resolving indirect anaphora and the outcome of the algorithm can be used for resolving all kinds of Anaphora (e.g. one anaphora, pronominal anaphora etc.)

1 INTRODUCTION

Resolving indirect anaphora is challenging because one try to encapsulate the unstated or background knowledge in the text. With previous studies researchers try to capture such knowledge by maintaining a tree structure starting from root and expands according to the decision of ‘has’ and ‘is part of’. Maintaining such a semantic tree having proper relation is essential in many problems [13, 14, 15, 17] but these algorithm limit to capture the hidden information of only one entity and later it becomes difficult to relate this information with appropriate referring expression and this referring expression has importance in answer extraction.

In this paper, proposed algorithm for identifying the difference in two co-refers (which are related indirectly) by calculating distance between these co-refer to root tree node (referring expression). For this an m-way tree is used and this tree expands at the same time it finds any entity attached with its node. So, the focus is on finding distance of a matched referent with its root node. After finding this distance we will compare it with another tree. Now the best suitable referent can be chosen for provided referring expression and available hidden knowledge. This proposed algorithm store distance in form of positive integers ranges from 0 to +∞. Hence, this algorithm is fast in comparing the outcomes coming from different tree traverse. Part of the algorithm falls in the category of solving the tree traverse problem.

2 RELATED WORK

Indirect anaphora, also known as bridging reference or associative anaphora, arises “when a reference becomes part of the hearer’s or reader’s knowledge indirectly rather than by direct mention” [3]. The object that is being referred to is called the anchor or the antecedent, the expression that refers to the antecedent is called the referring expression, and the association between the referring expression and the anchor is called the link. For example, the following sentence contains an instance of indirect anaphora [2].

First he go to the kitchen and then to the bedroom, the fridge door was open.

Unlike other types of anaphora, which can often be resolved using syntactic features, the resolution of indirect anaphora requires semantic knowledge of the relationship between the referring expression and the antecedent. Because such knowledge was previously unavailable to computer programs, most of the early studies in indirect anaphora were theoretical [4, 1]. In example, the referring expression, the fridge door, relates to the antecedent, the kitchen, but not to the bedroom through a whole/part (metonymy) link [5, 6]. For the question shown below, the indirect relation between fridge door and kitchen should be captured from text.
This approach not only returns the antecedent, but it can also reveal the type of association between each referring expression and its antecedent, which is a piece of information important for other parts of a full natural language processing system [2]. However, many frequently used types of links, such as event/role or cause/consequence, cannot be discovered by these systems because WordNet does not contain such knowledge [2]. There have been many successful machine learning based co-reference resolution systems [7, 8, 9], however most of them do not resolve indirect anaphora. Those do [8, 10] typically use the web as the corpus. Instead of searching through WordNet, they issue a series of web search queries made of the referring expression and each candidate antecedent [2]. So there are several web pages available over internet which are using the information of both referring expression and the candidate antecedent and both are used to measure the association strength. This association strength is used to measure the difference from some threshold value and if this difference is positive then the antecedent is supposed to be the exact antecedent for the referring expression. Here Machine learning techniques are used to determine the suitable threshold and so comparing the threshold with positive result values. Using this Machine learning technique by comparing threshold value provides better results comparable to WordNet implemented systems [11, 12, 13]. But a limitation of using this approach is that it does not determine the nature of semantic relation between referring expression and the antecedent. A more recent study of indirect anaphora has shown that precision for either approach can be significantly improved with a more sophisticated selection mechanism that learns by combining several features, such as salience [9, 16].

3 Proposed Algorithm

Step 1:
Take a referring expression say fridge door and construct a tree of all possible referents. Assign the value to \( \alpha \) according to predefined rules. Starting from \( \alpha = 0 \). Shown in Figure 1.

\[
\begin{align*}
\text{fridge door (it)} & : \alpha = -1 \\
\text{he} & : \alpha = 0 \\
\text{kitchen} & : \alpha = 0 \\
\text{bedroom} & : \alpha = 0
\end{align*}
\]

Figure 1: Referring expression relation for fridge door

Step 2:
Now traverse tree nodes and assign value \( +\alpha \) to those leaf nodes having \( -\alpha \) values in previous step. \( -\alpha \) value of any node in previous step indicates no use of the node for this calculation. And assigning \( +\alpha \) to \( \alpha \) indicates that this node is more expandable and can be used for different calculations. For value of \( \alpha \geq 0 \) expend the node further Shown in Figure 2.

\[
\begin{align*}
\text{fridge door (it)} & : \alpha = -1 \\
\text{he} & : \alpha = 0 \\
\text{kitchen} & : \alpha = 0 \\
\text{bedroom} & : \alpha = \infty
\end{align*}
\]

Conditions

\[
\begin{align*}
\text{kitchen} & \text{(expanded)} \\
\text{bedroom} & \text{(expanded)}
\end{align*}
\]

Figure 2: Referring expression relation for fridge door expended

Step 3(a): After tree nodes and assign value \( +\alpha \) to those leaf nodes having \( -\alpha \) values in previous step. \( -\alpha \) value of any node in previous step indicates no use of the node for this calculation. And assigning \( +\alpha \) to \( \alpha \) indicates that this node is more expandable and can be used for different calculations (fig. 3).
Now traverse tree nodes and assign value $+\infty$ to those leaf nodes having $-ve$ values in previous step. $-ve$ value of any node in previous step indicates no use of the node for this calculation. And assigning $+ve$ to $\infty$ indicates that this node is more expandable and can be used for different calculations.

**Fridge door (parent node/ root node)**

- kitchen (expanded) $\alpha = 0$
- stove $\alpha = +1$
- handle $\alpha = +2$
- fridge $\alpha = +1$
- vegetables

**fridge door**

$\alpha = +2$

matched for the value of $\alpha = +2$

**Figure 3: Referring expression relation for fridge door in kitchen**

**Step 3(b):** Take a referring expression say *fridge door* and construct a tree of all possible referents. Assign the value to $\alpha$ according to predefined rules. Starting from $\alpha = 0$. Shown in Figure 4.

**fridge door (parent node/ root node)**

- bedroom (expanded) $\alpha = 0$
- bed $\alpha = +1$
- handle $\alpha = +2$
- almirah $\alpha = +1$
- clothes

**almirah door**

**Figure 4: Referring expression relation for fridge door in bedroom**

**Step 4:**

**Compare:** Now comparing different values of step 3(a), Step 3(b), Step 3(c)… nodes and assign value $+\infty$ to those leaf nodes having $-ve$ values in previous step. $-ve$ value of any node in previous step indicates no use of the node for this calculation.
Algorithm to Calculate Semantic Distance among Indirect Question Referents

Step 5:
**Classification:** Classify according to the value $\mathcal{a}$.
Category 1: As $\mathcal{a} = \infty$ should be places in the table that will be least used for further discussions.
Category 2: Table of most promising indirect referents with value of $\mathcal{a} = +1$ and $\mathcal{a} = +2$.

<table>
<thead>
<tr>
<th>Sentence tag</th>
<th>Word tag</th>
<th>Referent Distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>St1</td>
<td>Wt1: he</td>
<td>$\mathcal{a} = \infty$</td>
</tr>
<tr>
<td>St1</td>
<td>Wt2: kitchen</td>
<td>$\mathcal{a} = +2$</td>
</tr>
<tr>
<td>St1</td>
<td>Wt3: bedroom</td>
<td>$\mathcal{a} = -\infty$</td>
</tr>
<tr>
<td>St2</td>
<td>Wt1: fridge</td>
<td>-</td>
</tr>
<tr>
<td>St2</td>
<td>Wt2: door</td>
<td>-</td>
</tr>
</tbody>
</table>

Step 6:
**Decision:** Now on basic of the value of $\mathcal{a}$, nodes and assign value $+\infty$ to those leaf nodes having $-ve$ values in previous step. $-ve$ value of any node in previous step indicates no use of the node for the calculation shown in table I.

**How this algorithm works:**
Large value of $\mathcal{a}$ is the indication of more promising referent and it is shown in figure 5 that larger the value of $\mathcal{a}$ more will be the chance to stay in the bucket of prominent referents.

![Figure 5: Weight of possible prominent referents](image)

Behavior of the $\mathcal{a}$ is shown in Figure 6, which shows that an entity having $\mathcal{a} = -\infty$ have no chances to be the prominent referent of the question.

![Figure 6: How the value of $\mathcal{a}$ affects the possible prominent referents](image)

Algorithm is calculating semantic distance for multiple referents in a single discourse for indirect anaphora used previously in the any of the Queue_Noun or Queue_Verb. It also extracts the ongoing discourse and semantic relation with the help of discussed parallel queue method. Extracted value of ongoing discourse and semantic relation can be used to identify indirect anaphora in same sentence. We aimed to present a general model for software estimation.
Our model concentrated on the pre-processing fuzzy inference system. The model was trained properly which was portrayed by the overwhelming training results. It was found out that Fuzzy Systems are better effort estimators than the previous methods. More work on hybrid models to resolve indirect anaphora need to be done with using a wide variety of parameters.

4 CONCLUSION AND FUTURE WORK

Previously parallel queue method have been proposed for resolving indirect anaphora as a new approach to save a very high value of time during long conversations. One major problem that facing while using this approach was the complexity of its implementation through programming and so implementation of this method requires a very high programming skill. As the future extension of the work, for the purpose of concluding theme which seems very easy part of this method as all Queues are storing information together attaching the tree information together. For better advancement of this work in future universal queue with the web instead of WordNet dictionary.

REFERENCES


Authors Profile

Dr. Namita Mittal is working as Assistant Professor since 1996, at Department of Computer Science and Engineering. She is recipient of Career Award for young Teachers. She is involved in teaching under graduate and graduate courses Database management System, Programming in Java, Information Retrieval, Data mining, Semantic Web etc. She has published several research papers in reputed international conferences and journals. She is also a member of review committees for Referred journals/conferences.

Lokesh Kumar Sharma has completed his BTech in computer science from Govt. Engineering College, Ajmer, Rajasthan. He has completed his MTech in ICT from Gautam Buddha University, Greater Noida, UP. Currently he is pursuing his PhD under supervision of Dr. Namita Mittal from Malaviya National Institute of Technology, Jaipur, Rajasthan, India.