ISSN No. 0976-5697

Volume 4, No. 2, Jan-Feb 2013

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

Indirect Anaphora Resolution using Parallel Queues

Lokesh Kumar Sharma*
School of Information & Communication Technology
Gautam Buddha University, Greater Noida
Uttar Pradesh, India
lokesh.gbu@gmail.com

Dr. Ela Kumar
School of Information & Communication Technology
Gautam Buddha University, Greater Noida
Uttar Pradesh, India
ela_kumar@rediffmail.com

*Abstract***:** When we are dealing with natural language text we come across the term Anaphora and for capturing the knowledge encoded in text Anaphora resolution is very essential. Indirect Anaphora are especially challenging to resolve because the referring expression and the antecedent are related by unstated background knowledge. So Such kind of Anaphora need to be resolved properly to capture the knowledge expressed in natural language. There are several ways to resolve Indirect Anaphora some of these treating as a different problem or resolving the Anaphora as semantic path search. In this paper we are proposing slightly different approach to deal with natural language inputs. And so, here first we are tagging lexically different natural inputs and putting these tagged inputs in some predefined parallel queues. Later using parallel queue communication we can determine ongoing discourse and semantic difference which is useful for Indirect Anaphora resolution.

Keywords: Anaphora, Indirect Anaphora, Universal Queue, Queue_Noun_Recency, Message Passing.

I. INTRODUCTION

Among several Anaphora discussed in language theory Indirect anaphora is a type of Anaphora that requires background knowledge in order to identify the referent. It may account for 15% of noun phrase anaphora making it an important type [1]. So Resolving Indirect anaphora is a necessary step for automatically capturing knowledge from text. However resolving indirect anaphora is problematic for cursory processing systems because it usually requires common sense knowledge. Because most knowledge sources contain little common sense knowledge, achieving a high level of recall in resolving indirect anaphora is especially difficult [2].

In this paper we are proposing a new approach for capturing ongoing discourse and semantic relations. As in the previous studies where relation between two entities are found by the semantic relation between them. This proposed system is very reliable in holding the discourse as it is using dominated queues after lexical tagging.

II. 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].

"When he go to ***the kitchen*** this time, *the door* was open." Here, the referring expression, *the door*, relates to the antecedent, ***the kitchen***, through a whole/part (metonymy) link.

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]. These studies identified a variety of types of indirect anaphora (see table I) [2].

Table I. Some frequent types of indirect anaphora [2].

Sr. No.	Link Type	Example
1	Set/element	a ***class***... the *student*...
2	Metonymy	a ***room***... the *wall*...
3	Hypernym/hyponym	an ***oak***... the *tree*...
4	Event/role	a ***murder***... the *killer*...

Recently there has been more progress in experimental studies of indirect anaphora. These studies can be divided into WordNet-based systems and machine learning systems [2]. The WordNet-based systems [5] use WordNet as the knowledge base. They take in a referring expression and a list of nouns that appear earlier than the referring expression in the same text. The systems choose one noun as the most likely antecedent for the referring expression. They select the antecedent by first grouping the nouns based on sentence boundary, then using stack-based theory [6] to sort the candidate associations and select the most promising one. Specifically, the systems look back one sentence at a time and return a candidate as the antecedent as soon as the candidate satisfies one of the following conditions [2] based on WordNet knowledge:

a. The candidate is a synonym of the referring expression, such as *aviator* and *flyer*.
b. The candidate is a hypernym (superclass) or hyponym (subclass) of the referring expression, such as *oak* and *tree*.

c. The candidate is a coordinate sibling of the referring expression, such as *home* and *house*.
d. The candidate has a meronymic (has-part) or holonymic (is-part-of) relation with the referring expression, such as *room* and *wall*.

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 coreference resolution systems, such as [7, 8, 9], however most of them do not resolve indirect anaphora. The ones that do [10, 8] 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.

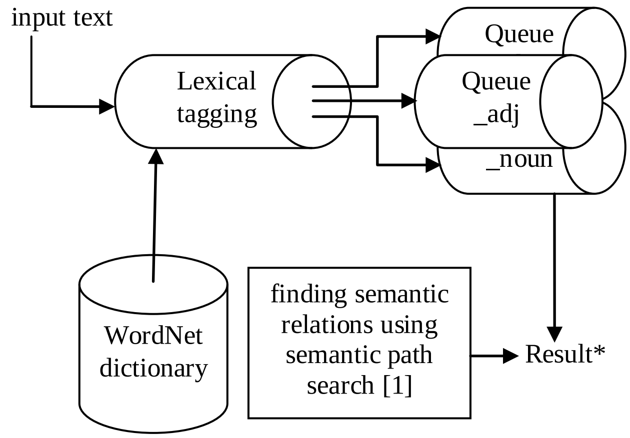
This association strength is used to measure the difference from some threshold value and if this difference is positive then the antecedent is supposed to be the exact antecedent for the referring expression. Here Machine learning techniques are used to determine the suitable threshold and so comparing the threshold with positive result values. Using this Machine learning technique by comparing threshold value provides better results comparable to WordNet implemented systems. 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 (the contextual distance of a referring expression and its anchor) and lexical distance (the semantic distance between a referring expression and its anchor) [11].

III. OUR SYSTEM

Our approach is based on the interconnections of various facts such as tagging the input in various part of speech and then developing a computer program that extracts the lexical meaning from the WordNet. For our example,

"When he go to ***the kitchen*** this time, *the door* was open".

We can treat this example as an input and putting this input in the tagging queue for lexical tagging that is gathering information from the WordNet dictionary. These tagged inputs are collected by dominating queues in next step. After this step queues work together in parallel. Gathering information from these parallel queues and finding relation from semantic path search algorithms discussed in previous studies. Figure 1 is showing the overall working of our proposed approach. This system is calculating ongoing discourse and semantic relation later helpful in Indirect Anaphora resolution.



*ongoing discourse and semantic relation.

Figure: 1 Overall diagram of parallel queue system for resolving Indirect Anaphora.

Now, we can explain the functioning of each of the parts of this proposed system as follow.

A. *Initial Gathering And Lexical Tagging:*

Now Here in the Figure 2 input text is passed through a universal queue** that consults with WordNet and with the help of this lexical dictionary universal queue collects the semantic information about all input text.

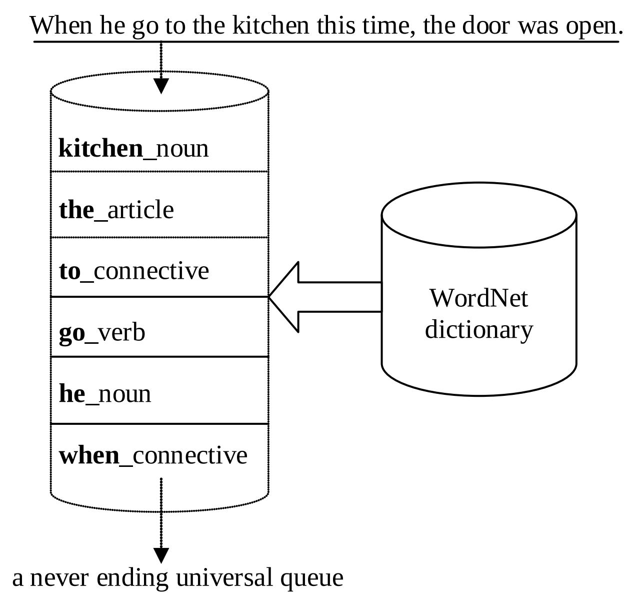


Figure: 2 Connection of Universal Queue with WordNet.

**universal queue is the very first infinite length queue which holds all of the input text together and keeps the semantic information with it.

B. *Dominated Queues For Semantically Different Inputs:*

After getting the semantic information about each text in the input we are ready to dominate a particular queue for similarly tagged inputs.

Here in Figure 3 these dominating queues are working parallel and collecting the semantically tagged inputs in an ongoing programmed queue. The benefit of this parallel queue structure is in further discussion.

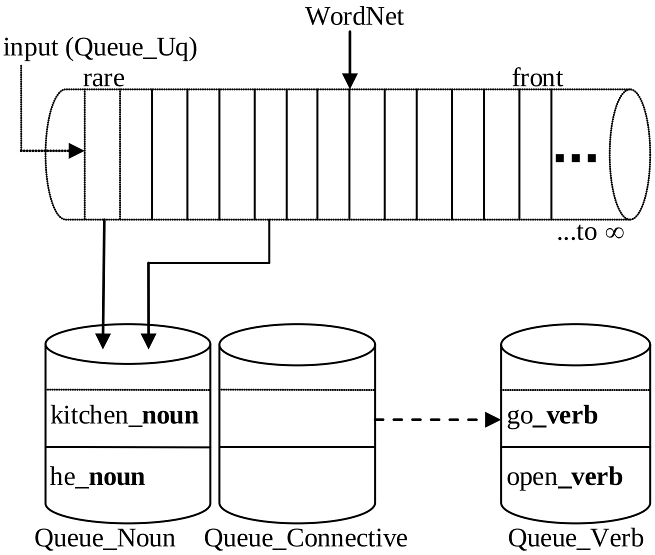


Figure 3: Semantic tagging of similar inputs in universal queue and putting them in to the dominating queues.

C. Ongoing Discourse message passing with the help of parallel queue structure:

Now, in this section according to Figure 4 we are moving forward to see the way of communication of two different parallel queues. First queue is storing the Noun and adjacent queue is storing the Verb and so these two can communicate simultaneously with the help of our computer program. Suppose any time Queue_Noun has its front on *kitchen* and rare on *he* and these both are noun when at the same time with in the paragraph *P1* we are looking at the front and rare of Queue_Verb it seems to be go and open.

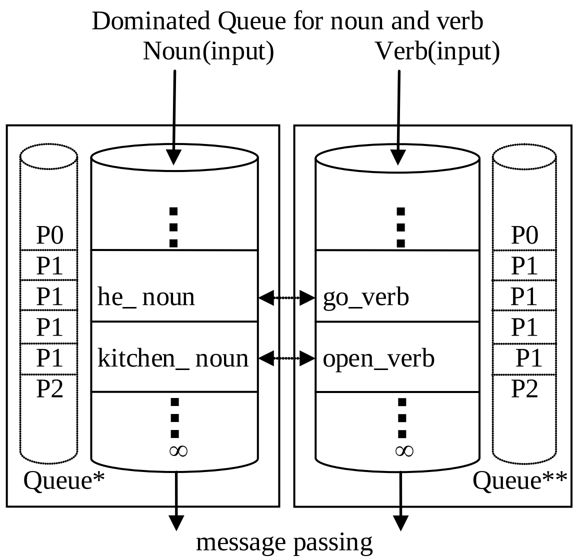


Figure 4: Parallel Queue communication and Storing information in adjacent Queue.

*Queue_Noun_Recency: It is another queue along with primary Noun queue which stores the information about the running paragraph Noun associated with tree data structure.

**Queue_Verb_Recency: Similarly, it stores the information about the running paragraph Verb and associated with tree like data structure.

So, in this case we collect the information of regarding queue from the Queue_Noun_Recency*. So, *QNR plays an important role here. It is mentioned as follow,

Kitchen ***has part*** doors. Door ***has part*** lock.
Kitchen ***has part*** windows. Window ***has part*** bolt.
Kitchen ***has part*** walls. Wall is ***part of*** building/room/house.

Kitchen may ***have*** a fridge, stove, lighter etc..

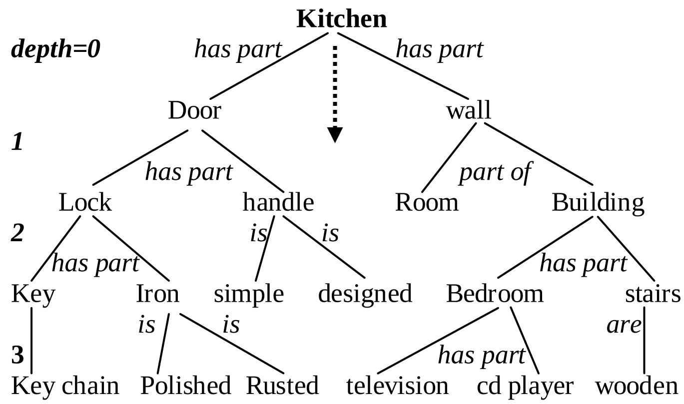


Figure 5: Queue_Noun_Recency is storing information in the form of tree structure which is bound to the depth of 3.

QNR can store these information in tree like structure shown in Figure 5 where the information of ongoing noun is in the form of *has part* and *part of* and so it is going to be deep storage for the purpose of understanding, but here we are only interested in capturing the indirect meaning of the ongoing noun so we can bound our storage at certain level that may also follow any rule. For a simple example lets bound the Queue QNR with the condition that in a tree structure it can go up to depth=3.

D. Context at any time of conversation, at time 't':

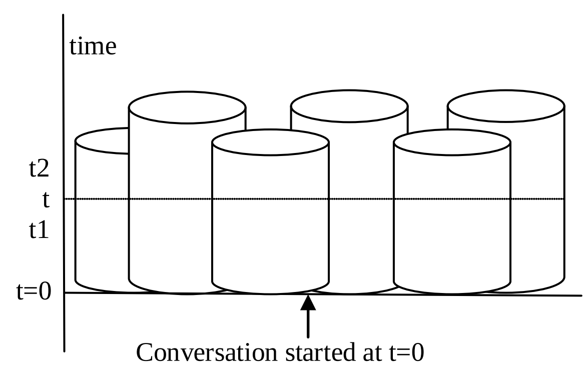


Figure 6: Capturing context of discourse at any time '*t*'.

Now, here we can show that how we are able to extract an important conclusion from this parallel queue method. Since the starting of the dominated parallel queues the discourse is running parallel and storing the result and information in parallel. So, at any time of conversation is now very easy and very fast to switch over any conversation context at any time *t*.

In previous studies [12, 13, 14, 15, 16, 17, 18] we were not able to point on ongoing discourse at any time ‘t’. But here after considering these dominated parallel queues we can point at any time and also can capture discourse for any time interval.

IV. RESULT

So, finally we have an interface shown in Figure 7 which is telling us about the Indirect Anaphora used in the any of the Queue_Noun or Queue_Verb.

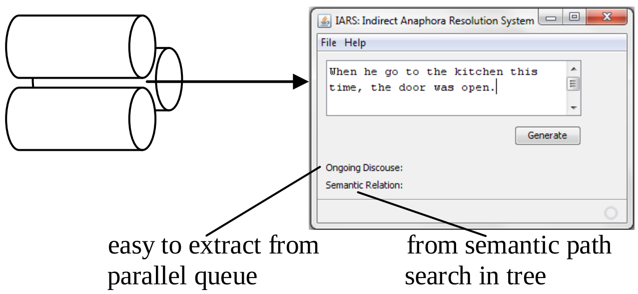


Figure 7: Implemented Interface for showing results.

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 identified Indirect Anaphora in same sentence.

V. CONCLUSION AND FUTURE WORK

We proposed Parallel Queue method for resolving indirect anaphora as a new approach to save a very high value of time during long conversations. One major problem that we face while using this approach was the complexity of its implementation through programming and so implementation of this method requires a very high programming skill.

In future we can extend this work for the purpose of concluding theme which seems very easy part of this method as all Queues are storing information together. For better advancement of this work in future we can also attach the Universal Queue with the web instead of WordNet dictionary. Here in Figure 5 we show a tree with bounded level that can be extended further for finding semantic distance itself.

VI. REFERENCES

[1] M. Poesio and R. Vieira. A corpus-based investigation of definite description use. Computational Linguistics, 24(2):183–216, 1998.

[2] James Fan, Ken Barker, and Bruce Porter. “Indirect Anaphora Resolution as Semantic Path Search”, In Proceedings of Third International Conference on Knowledge Capture, Canada. October 2005.

[3] R. Mitkov. Anaphora Resolution. Longman, London, 2002.

[4] C. Gardent, H. Manuelian, and E. Kow. Which bridges for bridging definite descriptions? In Proceedings of the 4th International Workshop on Linguistically Interpreted Corpora, 2003.

[5] R. Vieira and M. Poesio. An empirically based system for processing definite descriptions. Computational Linguistics, 26(4):539–593, 2000.

[6] C. L. Sidner. Towards a computational theory of definite anaphora comprehension in English discourse. PhD thesis, MIT, Cambridge, MA, 1979.

[7] Wee Meng Soon, Hwee Tou Ng, and Chung Yong Lim. 2001. A machine learning approach to coreference resolution of noun phrases. Computational Linguistics, 27(4):521–544.

[8] W. M. Soon, H. T. Ng, and D. C. Y. Lim. A machine learning approach to coreference resolution of noun phrases. Computational Linguistics, 27(4):521–544, 2001.

[9] Vincent Ng and Claire Cardie. 2002. Improving machine learning approaches to coreference resolution. In Proc. of ACL, pages 104–111.

[10] K. Markert, M. Nissim, and N. Modjeska. Using the web for nominal anaphora resolution. In Proceedings of the EACL Workshop on the Computational Treatment of Anaphora, pages 39–46, 2003.

[11] M. Poesio, R. Mehta, A. Maroudas, and J. Hitzeman. Learning to resolve bridging references. In Proceedings of ACL 2004, 2004.

[12] R. D. Burke, K. J. Hammond, V. Kulyukin, N. T. Steven L. Lytinen, and S. Schoenberg. Question answering from frequently asked question files: experiences with the FAQ FINDER system. AI Magazine, 18(2):57–65, 1997.

[13] S. E. Preece. A spreading activation model for information retrieval. PhD thesis, University of Illinois, Urbana Champaign, USA, 1981.

[14] D. L. Waltz and J. B. Pollack. Massively parallel parsing: A strongly interactive model of natural language interpretation. Cognitive Science, 9:51–74, 1985.

[15] R. Kjeldsen and P. Cohen. The evolution and performance of the GRANT system. IEEE Expert, 2(2):73–79, 1987

[16] L. Rau. Knowledge organization and access in a conceptual information system. Information Processing & Management, 23(4):269–283, 1987.

[17] B. Onyshkevych and S. Nirenburg. A lexicon for knowledge-based mt. Machine Translation, 10(1-2):5–57, 1995.

[18] P. Cohen and R. Kjeldsen. Information retrieval by constrained spread activation on semantic networks. Information processing & management, 23(4):255–268, 1987.