Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X
IMPACT FACTOR: 6.017

IJCSMC, Vol. 6, Issue. 3, March 2017, pg.131 – 137

Efficient Methodology to Improve Web-Based Searches

Sai Kishan T, Rajasekar G

Rajalakshmi Engineering College, India

saikishan95.t@gmail.com, rajsekar95@gmail.com

Abstract— This paper presents a plausible solution to resolving the issue of one aspect of Anaphora in searches that provides better results for your search. Using simple parsers and POS tagging, we solve the problem of Anaphora. This algorithm can even narrow your search to one specific result. The algorithm has been implemented partially and so it is difficult as of now to talk about its efficiency in terms of performance. But its working can be traced hypothetically and see how it solves the queries presented to it.

Keywords— Anaphora resolution, POS tagging, Book suggestion.

I. INTRODUCTION

Anaphora poses a very serious threat to a variety of domains of research and applications in the real world. It is one thing for humans to process the words of a language known to them, but for a machine to work out the intricacies of the same language is something infinitely more complex. With this algorithm, we look forward to solving a simple form of anaphora that occurs in our day to day use of words and language.

Anaphora is associated with the Greek word *anajora*, *Ana* meaning backward or upward and *jora* meaning the act of carrying back upstream. In linguistics, anaphora is the use of an expression whose interpretation depends upon another expression in context (its antecedent or postcedent). In a narrower sense, anaphora is the use of an expression that depends specifically upon an antecedent expression and thus is contrasted with cataphora, which is the use of an expression that depends upon a postcedent

expression. The anaphoric (referring) term is called an anaphor. For example, in the sentence *Sally arrived, but nobody saw her*, the pronoun *her* is an anaphor, referring back to the antecedent *Sally*. In the sentence, *before her arrival, nobody saw Sally*, the pronoun *her* refers forward to the postcedent *Sally*, so *her* is now a *cataphor* (and an anaphor in the broader, but not the narrower, sense). Usually, an anaphoric expression is a proform or some other kind of deictic (contextually-dependent) expression. Both anaphora and cataphora are species of endophora, referring to something mentioned elsewhere in a dialog or text.

In the following example, 1) and 2) are utterances; and together, they form a discourse.
1) John helped Mary. 2) He was kind.

As human, readers and listeners can quickly and unconsciously work out that the pronoun "he" in utterance 2) refers to "John" in 1). The underlying process of how this is done is yet unclear, especially when we encounter more complex sentences.

An example involving Noun phrases;

1a) John travelled around France twice.
1b) they were both wonderful. ??

2a) John took two trips around France.
2b) they were both wonderful.

Consequently, anaphora resolution presents a challenge, and is an active area of research. Through this algorithm of ours, we intend to solve this kind of an Anaphora for sentences having simple proper nouns as their subject or object. With simple tweaks to the actual algorithm, it could probably be used to solve a few more complex occurrences of Anaphora.

When it comes to literature, there are a lot of forums for book reviews and comments. There are even a few good domains that suggest books based on the given criteria. The extents of the criterion are narrowed down to very small constraints like genre and name of the author and the suggestions for the given criterion consist of a wide range of books. So by using the concepts of POS tagging and anaphora resolution, we intend to narrow this search down to a very minimal list of books thereby, helping the user come down to only handful of choices or sometimes even one specific book of the author's prescribed criterion.

This site an application project in Artificial intelligence domain. In specific it is an expert system that can give you better results for your search. There are umpteen number of sites which host blogs and pages for books and reviews. There are also sites that give you the list of books you are searching for. This application project can even narrow your search to one specific book. This site uses anaphora resolution algorithm for narrow search results. This site is also for readers who are yet to inculcate the reading habit and don't have any idea where to begin. This application can be used to review, create pages and share information about any literature related event. This can be used to create a direct network between authors and readers. It is an improved search feature that uses language processing, in order to give a refined result. This search can also function as an expert system, which suggests books suited for your tastes and choices. The advantages of the system are refined search results, a forum for readers and authors, completely User- driven, not only for ardent book readers, but also for beginners.

II. LITERATURE SURVEY

An algorithm for pronominal anaphora resolution by Shalom Lapin & Herbert J. Leass;

1) A morphological filter for ruling out anaphoric dependence of a pronoun on an NP due to non-agreement of person, number, or gender features. The filter works based on the following conditions;

A pronoun P is non-co referential with a (non-reflexive or non-reciprocal) noun phrase N if any of the following conditions hold:

   a) P and N have incompatible agreement features.
   b) P is in the argument domain of N.
   c) P is in the adjunct domain of N.
   d) P is an argument of a head H, N is not a pronoun, and N is contained in H.
   e) P is in the NP domain of N.
   f) P is a determiner of a noun Q, and N is contained in Q.

2) A procedure for identifying pleonastic (semantically empty) pronouns
3) An anaphor binding algorithm for identifying the possible antecedent binder of a lexical anaphor (reciprocal or reflexive pronoun) within the same sentence (This algorithm is presented in Lapin and McCord 1990b.)
4) A procedure for assigning values to several salience parameters (grammatical role, parallelism of grammatical roles, and frequency of mention, proximity, and sentence recency) for an NP. (Earlier versions of these procedures are presented in Leass and Schwall 1991.) This procedure employs a grammatical role hierarchy according to which the evaluation rules assign higher salience weights to (i) subject over non-subject NPs, (ii) direct objects over other complements, (iii) arguments of a verb over adjuncts and objects of prepositional phrase (PP) adjuncts of the verb, and head nouns over complements of head nouns. 1
5) A procedure for identifying anaphorically linked NPs as an equivalence class for which a global salience value is computed as the sum of the salience values of its elements.
6) A decision procedure for selecting the preferred element of a list of antecedent candidates for a pronoun.

III. ANAPHORA RESOLUTION ALGORITHM (ARA)

This algorithm usually does simple POS tagging to the given input query and extracts only the subject and object from the given input. It works in multiple stages.

I. The first stage is POS tagging, where we remove all the unwanted parts of speech from the sentence and take only the subject and object.
II. We take only the subject and object, irrespective of them being a proper or a common noun; we search for them on the internet and generate a set of anaphora. For example, if your subject is the prime minister of India, we crawl through various web pages and generate a list of endophora like Narendra Modi, ex-chief minister of Punjab and so on. If it is just a proper noun like Jack or Mary, we just find out the gender of the subject or object and fill the list with suitable pronouns.
III. We then parse through the whole passage to see if there are any other subjects or objects mentioned other than those in the first sentence.
IV. If there are more subjects or objects, we add them to a stack in their order of occurrence.
V. We then parse through the passage, sentence by sentence and find out if there are two or more subjects and objects in that sentence.
VI. If there are two objects, we just parse through the list of anaphora generated online and resolve them.
VII. If there are more than two subjects and objects, we also make use of the stack we have created and resolve the anaphora by the order of their occurrence.

IV. ADVANCED SEARCH

We make use of the ARA in our project to narrow down our search results for our book review site. We have added a feature to our web application, with which it is possible to search for books in our database, without knowing the author or the title and just by a small description about the story. For this process, we have followed the following steps,

I. We first perform POS tagging on the query and obtain only the subject and object from the whole query passage.

II. Now using term frequency, we obtain the passages from various pages or from tables on our database that contain the subjects and/or objects in our passage.

III. We now feed the query and the passage into the ARA and obtain the anaphorically resolved output.

IV. Then we do a process called Verb Simplification. We simply remove the verb and replace it with the simple form of the verb. For example, if the verb is in the future or past tense or even in some continuous tense, we remove it and replace it with the root verb.

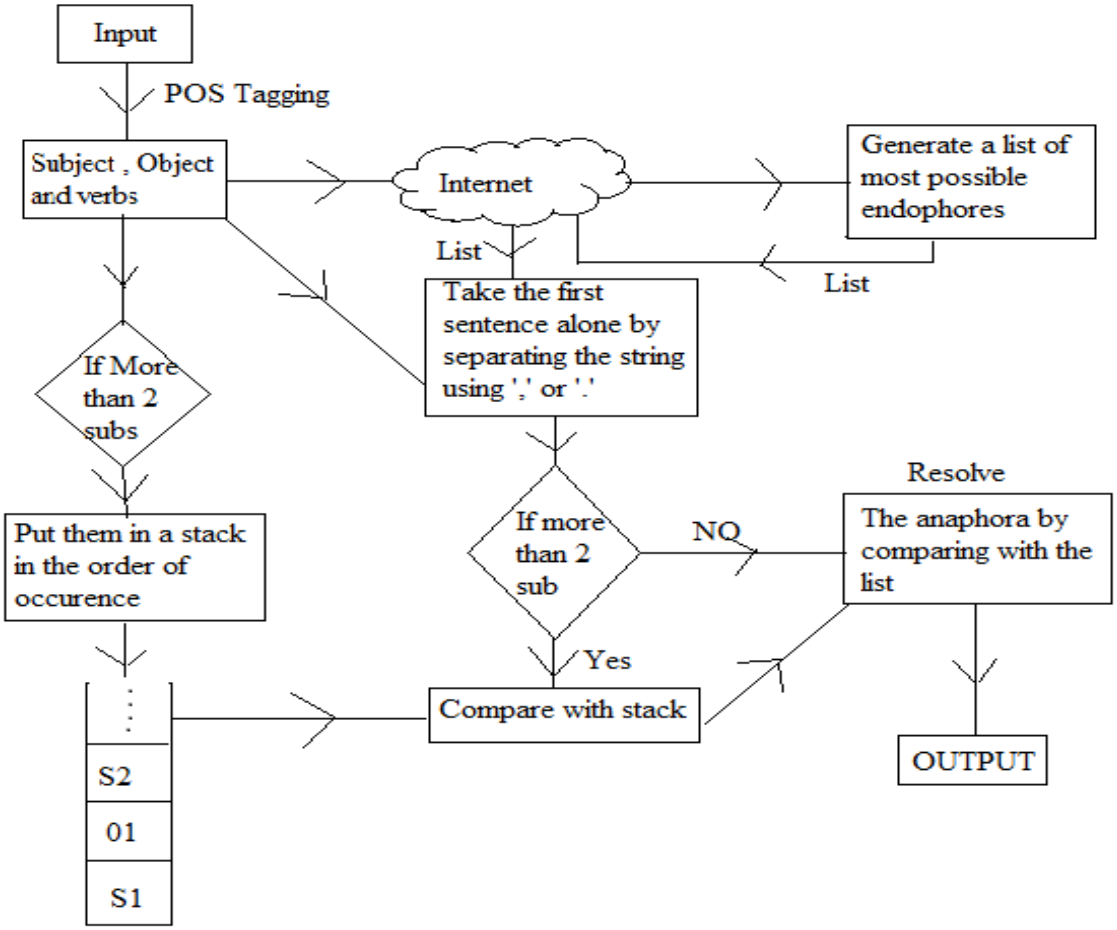
V. We perform verb simplification on the passage obtained as well as the query given by the user.

VI. Now we perform simple comparisons and find out if the query given by the user and the passage obtained are the same.

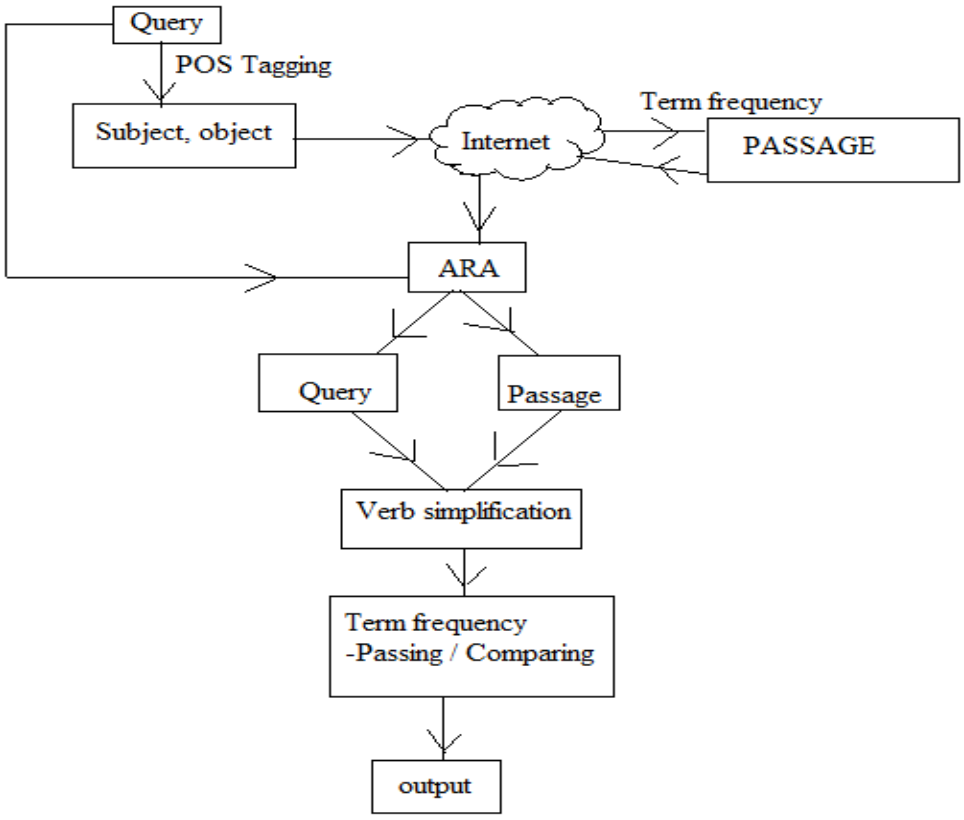
VII. Thus we obtain results based on simple comparisons.

V. DFD

I.

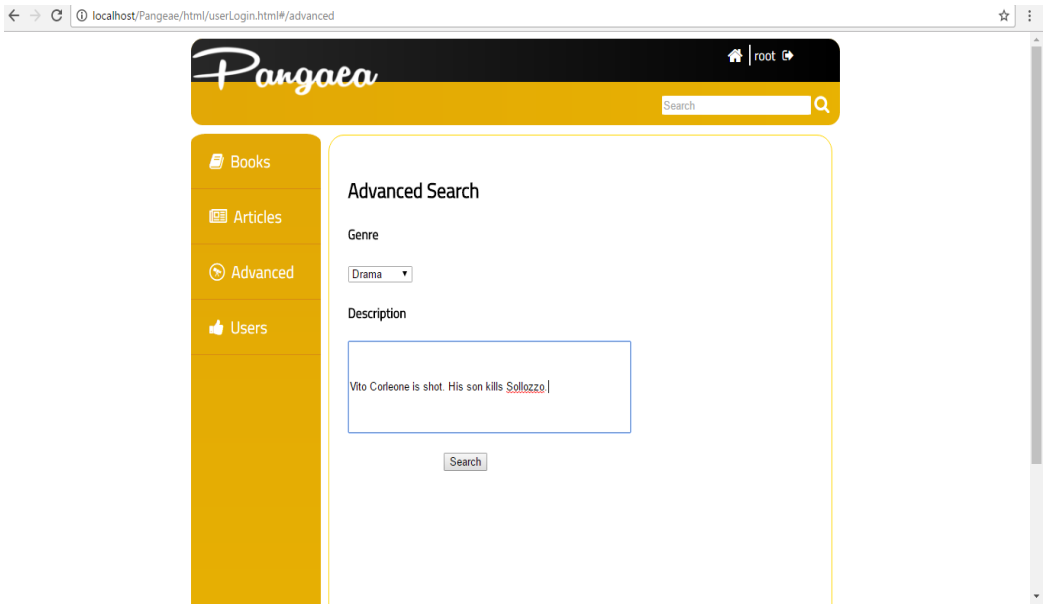


II.

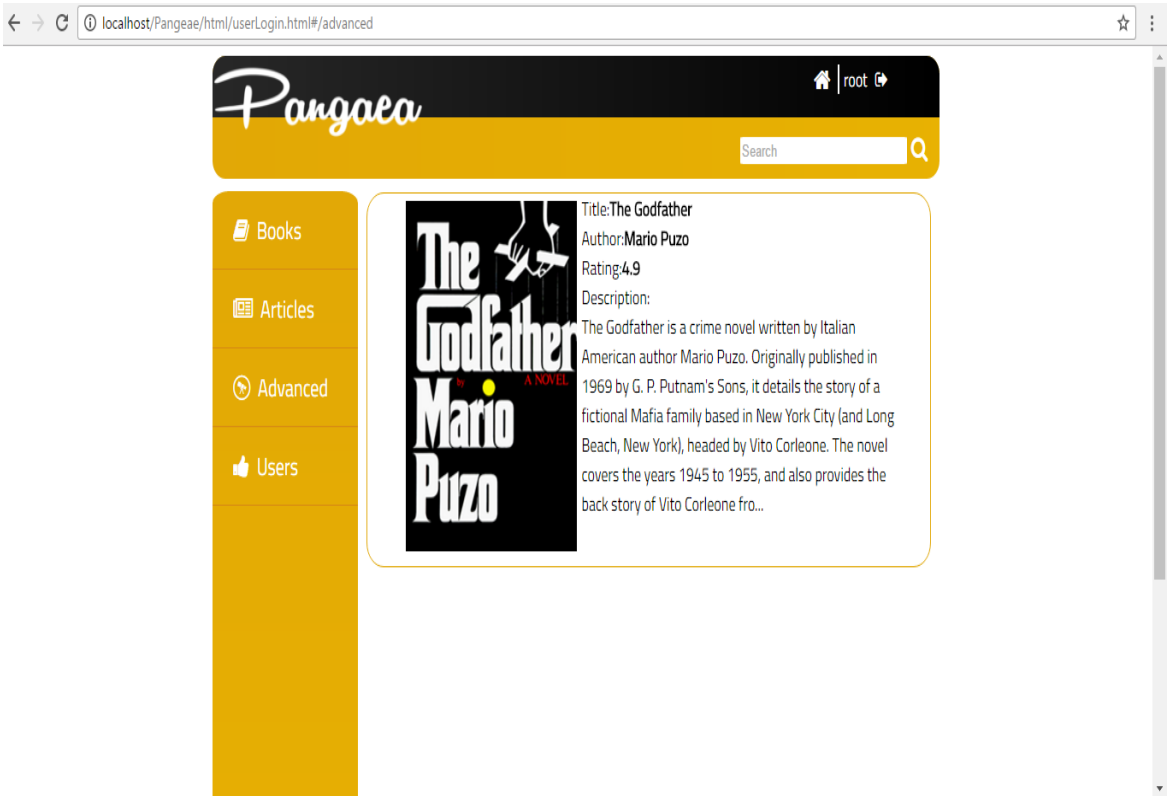


VI. WORKING SCREENSHOTS

I.



II.



VII. Future Works

- We have created a simple user subscription module, which allows users to subscribe and read the articles and blogs posted by other users. This can be further built and constructed into a fully functional social network for authors, readers and publishers to share their thoughts and reviews and to give budding writers a chance to prove their talents.
- The ARA can be further enhanced to solve direct tone and compound sentences having more than two or three sentences.
- This is a user driven application, and so when a user searches for a book and it doesn't already exist in the database, it is possible to create a module which searches for the same query online and automatically create a page for the book by simple web crawling.
- When a user enters a query, it is highly probable to have mistakes in the spelling of proper nouns and so results obtained will not be up to the mark. So, an algorithm could be used to generate results for either the query or something that is close to the query.

VIII. CONCLUSION

This algorithm, proposed to resolve Anaphora, can solve simple queries. If a compound sentence or a sentence involving a direct tone (A sentence like; Mary said, "He is…") would have some problems being correctly parsed. Results will be obtained, but the accuracy may be compromised. But simple sentences and a few compound sentences will get you good results.

The application as a whole will help you a great deal when you don't know the exact name of the book, but you know its plot. The advanced search which was built using the ARA will help you to narrow down to the book you have been searching for. It can also help you find books which fit to a description you have in your mind.

REFERENCES

- http://www.aclweb.org/anthology/J94-4002
- http://stackoverflow.com/questions/28618400/how-to-identify-the-subject-of-a-sentence
- http://stackoverflow.com/questions/30016904/determining-tense-of-a-sentence-python
- http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.228.7950&rep=rep1&type=pdf
- http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.52.597&rep=rep1&type=pdf
- https://pdfs.semanticscholar.org/26fd/249a167b09684b61650f99ad87d8358eeb0a.pdf

