A Discourse Linking Tool for English Language Texts comprising Lexicon building and Ontology Creation

Anuja Bawaskar
College of Engineering, Pune
Pune, Maharashtra, India
anujabawaskar@gmail.com

Kasturi Adep
College of Engineering, Pune
Pune, Maharashtra, India
adepkc13.comp@coep.ac.in

Adarsh Jaju
College of Engineering, Pune
Pune, Maharashtra, India
jajuar13.comp@coep.ac.in

Yashodhara Haribhakta
College of Engineering, Pune
Pune, Maharashtra, India
ybl.comp@coep.ac.in

Krishnanjan Bhattacharjee
Applied AI Group, CDAC
Pune, Maharashtra, India
krishnanjanb@cdac.in

Swati Mehta
Applied AI Group, CDAC
Pune, Maharashtra, India
swatim@cdac.in

Ajai Kumar
Applied AI Group, CDAC
Pune, Maharashtra, India
ajai@cdac.in

ABSTRACT

Natural Language text is not bound by a fixed structure. For a machine to understand the language, the challenge lies in resolving the ambiguities and capturing innovativeness. Due to its unstructured nature, discourse linking, required for understanding and generating text by a machine is a challenging task. Also, dealing with sentences varied in nature and changing them into generic structure is an additional challenge. This paper presents a way to create a discourse linking tool for English language. This tool is based on specially created lexicon and hand-crafted rules suited for discourse linking purpose. Currently available lexicons such as dictionaries, WordNet, etc. are not suited as they lack domain knowledge. Therefore, an ontology is developed for political news domain which serves as a lexicon and its inherent relations are used for discourse linking purpose.

CCS CONCEPTS

• **Computing Methodologies** → **Natural Language Processing**; • **Computing methodologies** → **Discourse, dialogue and pragmatics**; Information extraction;

KEYWORDS

Discourse linking, co-reference resolution, anaphora resolution, ontology, lexicon building, chunking.

ACM Reference format:
Anuja Bawaskar, Kasturi Adep, Adarsh Jaju, Yashodhara Haribhakta, Krishnanjan Bhattacharjee, Swati Mehta, and Ajai Kumar. 2017. A Discourse Linking Tool for English Language Texts comprising Lexicon building and Ontology Creation. In *Proceedings of International Conference on Computer and Communication Technology, Allahabad, India, November 24–26, 2017 (ICCCT-2017)*, 6 pages.
https://doi.org/10.1145/3154979.3154986

ICCCT-2017, November 24–26, 2017, Allahabad, India

ACM ISBN 978-1-4503-5324-3/17/11…$15.00
https://doi.org/10.1145/3154979.3154986

1 INTRODUCTION

Most of the data available today in the world is in a textual format, thus the need for processing and understanding of the text is fueled. Natural Language Generation through computers is one of the toughest challenge for Language Processing and solving entity-action relationship through anaphora and co-referencing is pertinent to the task. The linking of discourse is also much needed for computer generated text summary of large documents. In order to attain true cognition, language generation remains the key, hence, a workable solution towards discourse linking and NLG is a need of the hour in this digital era.

This paper presents a discourse linking tool conceived from English Language Grammatical and Logical relations in a discourse through an indigenous algorithm in conglomeration with few open source tools to identify the connections between chunks spanned across different sentences. The tool attempts to resolve relations between sentences by addressing the pronominal co- reference, aliases and anaphora found in textual data. This tool has been developed to facilitate Natural Language Generation, Automatic Text Summarization where through the discourse linking, cohesion problem of computer generated summary can be solved. Moreover, the linked discourse aids in developing an NLP based Question-Answer (QA) system capable of giving semantically more relevant and inclusive results than normal keyword based search and retrieval.

For complete understanding of a text, domain knowledge is necessary. Building a customized lexicon attempts to come up with a standardized platform to incorporate the field knowledge with high extensibility. Datasets, ontologies, training statistical models based on an expert-annotated corpus are proposed solutions for collecting and storing context dependent information. The created

software uses an ontology to represent domain knowledge required. An ontology provides a framework for storing hierarchical data and their inter-relationships. Ontologies can also be queried using different reasoners. Web Ontology Language (OWL) API provides interface for automatic creation and updating of ontologies.
For resolving pronouns and anaphora, a deterministic approach is used. The rule-set formulated and implemented in the tool can be extended to take into account complex structure sentences containing multiple clauses. Statistical approaches on the other hand require a large set of manually annotated corpora to train a substantially performing model which is highly tiring. Taking the pros and cons of both the methods into account, the world is moving towards combining the advantages of both these approaches and creating a hybrid implementation. The system described in the paper is a hybrid approach which uses deterministic techniques that depend on the grammatical structure of the English language along with an ontology based lexicon which can be populated using statistical ontological learning techniques.

2 RELATED WORK

Development of resources that identify interconnections between discourse structures has been of interest to move language technology beyond level of the sentence in the fields of Natural Language Generation (NLG), text summarization and question answering. For establishing complete understanding of a text, a system needs to identify discourse relations as well as co-reference relations.
In the field of discourse relations, the Penn Discourse Tree Bank(PDTB) and PropBanks ARGM[1] have worked more towards exploring discourse relations annotated by marking the necessary lexical items - called discourse connectives. The Discourse Relation annotation scheme is designed to capture the logical flow of a text and its structure by marking the so-called discourse connectives and their arguments, as well as some additional information regarding their meaning and attribution. However the problem of co-reference is not addressed.
The StanfordCorefAnnotator performs pronominal and nominal co-reference using different techniques like deterministic (fast rule based)[2], statistical (a machine learning approach) and neural (neural-network based)[3], of which the statistical approach gives maximum F1 measure. Out of these 3 models, only statistical and neural models are trainable. OrthoRef and ANNIE by GATE resolve orthographic co-reference and orthographic and pronominal co-reference respectively. Identification and extraction of different features for chunks of texts is a crucial step in creating a co-reference resolution system.[4] Using an existing tool for resolution requires manual separation of the feature extraction module in these systems and interfacing it with a self-designed system that implements further improvements[5]. This limited scope for improving the existing tools motivates us to create a platform with higher flexibility.

3 METHODOLOGY

This section explains the indigenous modules implemented to put the system together. Figures 1 and 2 depict different processes that are part of the system and their flow.

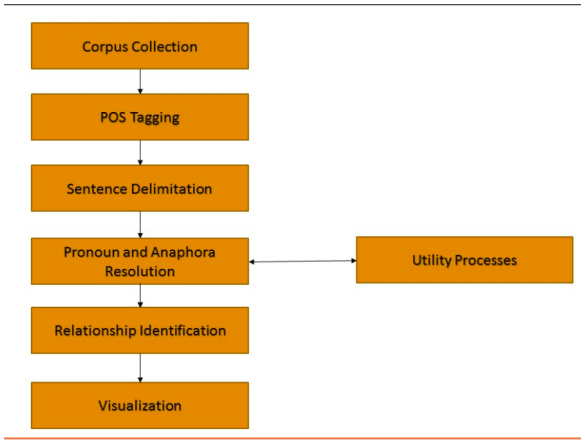

Figure 1: Process Flow for Discourse Linking Tool

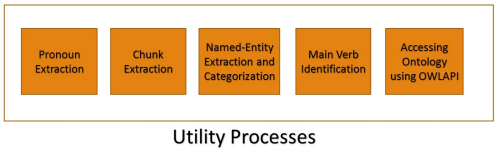

Figure 2: Utility Processes

3.1 Corpus Collection

A corpus consisting of 500 news articles from The Times of India (2016-2017) was identified. The said corpus was used for identifying specific patterns and relationships within and across sentences leading to the formulation of techniques for pronoun and anaphora resolution. The corpus collected, consisted of diverse news articles based on the length of the text and number of named entities, pronouns, anaphora and different ontological relationships present. Consider the following example of sentences.
e.g. Chief Minister Kamal Rao is visiting Pune. Chief Minister is planning for the city's development.
In the second sentence Chief Minister refers to Kamal Rao, this is an ontological relationship.

3.2 Sentence Delimitation

A period character can serve many functions - it can be used to denote the end of sentence, as a decimal point in mathematics, as a full stop after initials of names, in computing to denote IP addresses and websites, or in abbreviations. A sentence is a set of words that is complete in it itself and hence being able to break the document in all its constituent sentences is essential in constructing the meaning of the entire document. Moreover, the corpus used consists of news articles that contain quoted text. Text within quotes

may contain more than one sentences, but all of them have to be captured together to extract the complete sense of what is being said in the dialogue. The sentence delimitation module has been designed to fulfill the above purposes. This module outputs a list of delimited sentences that are further processed by the tool.

3.3 Utility Processes

There are five utility processes.

3.3.1 Pronoun Extraction. This module examines the POS tags of words in the text and identifies pronouns from it. A pronoun is uniquely identified with the POS tag, sentence number, word number with respect to the sentence. Apart from the aforementioned parameters a pronoun is associated with a 'coref' attribute that stores output of the pronoun resolution module and a 'chunkNECode' attribute that describes the resolved text. A list of pronouns is maintained throughout the processing cycle of the tool which also helps while evaluating the tool's performance.

3.3.2 Chunk Extraction. Chunk Extraction, the process of identification of chunks i.e. groups of words that when combined together provide more general or abstract information. Chunks are candidates used for pronoun resolution where the pronoun may be referencing the complete chunk under consideration or a part of the chunk.

The process of chunking is largely based on the POS tags of the given text. The module identifies noun phrases, noun phrases with conjunction and noun phrases with preposition phrases as chunks. The implemented algorithm for chunk identification is based on a finite state model, a part of which is shown in figure 3, where state transitions occur on the basis of input tokens, and a chunk is accepted on reaching the accept state.

Analogous to the pronouns of the application the chunks are also associated with certain attributes that qualify their occurrences in the text. These attributes are the grammatical number or plurality of a chunk (singular/plural), named entity category (person/location/organization) it is most likely associated with and the gender (male/female), animacy of the chunk. Singularity of a chunk depends on the type of phrase that is identified. If a chunk contains noun phrases connected through a conjunction it is defined as plural. However if one of these noun phrases is a named entity then the number of occurrences of named entities of that particular type govern the plurality. The default case looks at the POS tag of the last word in the chunk to determine the grammatical number of the chunk. For determination of NER category, the named entity that occurs last in the chunk is identified except in case where the chunk contains a preposition phrase associated. Next important parameter for chunks is the animacy of the chunk, i.e whether the chunk talks about an animate entity or an inanimate entity. For assigning animacy to a chunk the system first takes into consideration the NER category, if identified as a 'PERSON' the chunk is animate. The next step is to identify the key noun from the chunk and query the ontology. If the ontology does not contain an instance for that noun under the 'Animate' class we assign the value inanimate. The system provides flexibility to add new features such as sub-category that may be used for anaphora resolution. For example 'Samanya Jan Party' is a political party. The identified named entity will have an NER category of 'ORGANIZATION'. Using the ontology the named entity can further be classified as a political party. For further anaphoric references such as 'The political party' these features can be used.

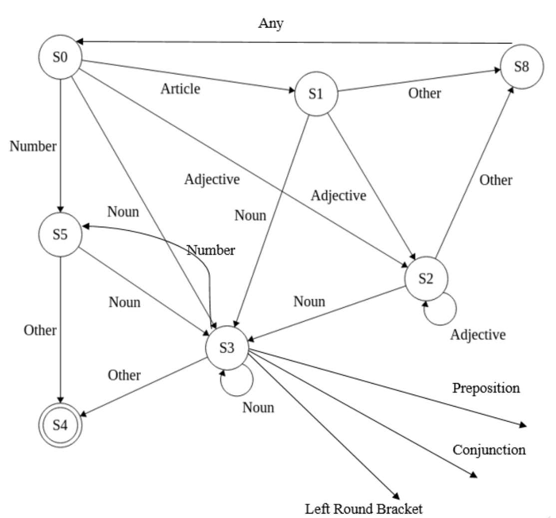

Figure 3: Chunk Extraction Process

3.3.3 Named Entity Recognition and Categorization. For Named Entity Recognition (NER) and Categorization the system uses Stanford's NER. Stanford's NER has high precision and recall values according to a study conducted[6]. It also provides user to train new models and gives the tool extensibility to identify named entities in a given domain. As context helps the NER identify and categorize named entities better, complete sentence text is given as an input to it. The output obtained is in the 'tabbed entities' format which is parsed to extract named entities and their categories. Named entities are qualified by attributes such as their category (PERSON, LOCATION, ORGANIZATION), gender (in case of PERSON), sentence number and word number.

3.3.4 Main Verb Identification. Identification of main verb is necessary to break a sentence in Subject-Verb-Object format which is used for pronominal co-reference. The tool uses the Stanford dependency parser for this process. The root relation gives the main verb of the sentence, or the cop relation gives the main verb of the subject in case of presence of a copular verb.

3.3.5 Accessing Ontology using OWL API. To store domain specific data we use an ontology in combination with owlapi-osgidistribution 3.5.6. The ontology functions as the customized lexicon. The ontology holds data which may be required to qualify the extracted named entities. The ontology stores different Indian names to determine the gender of a named entity categorized as PERSON by the Stanford NER. It stores a standard set of abbreviations along with their expansions, geographical locations and their interrelations, their aliases, list of animate and inanimate objects. It stores a list of animate nouns identified by analysis of new articles and

WordNet synsets for these nouns. The ontology also incorporates the hierarchy of the Indian political system. OWL API provides functions to populate an ontology using Java (creating classes, instances,relationship properties and data properties), to get the subclasses and super-classes of an instance, to check existence of an instance. For querying we use the HermiT reasoner. This ontology forms a major segment in the creation of a domain specific lexicon.

3.4 Pronoun Resolution

This section discusses the pronoun resolution approach implemented in the tool. For each of the approaches discussed below, the word number and the line number of the pronoun are maintained. The output for this is stored as an array of attributes namely the pronoun name, resolved string, and the sentence number
Flowchart presented in figure 4 depicts the algorithm for resolution of the pronouns of the classes he (he, him, his) and she (she, her, hers). This can be explained by the following example. *e.g. Kamal Rao who won from ward 31 said he was denied a ticket in the 2012 corporation polls. "I then joined the Youth Party in 2014 with my wife. This seat was never with the Youth Party for the last 20 years." he told. He won the seat this year. Congregation Party also denied party leader Vijay Tiwari his seat. The party leader then left Congregation Party and joined Jan Sena Dal. The Dal welcomed Tiwari and others with open arms.*

In the first sentence 'he' is in the object (main verb is said) and

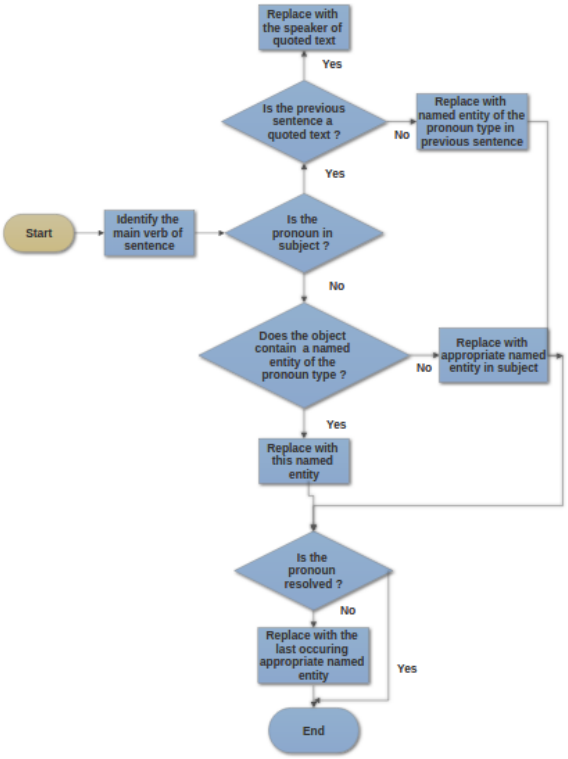

Figure 4: Resolution of pronouns of the classes he (he, him, his) and she (she, her, hers)

since there are no appropriate named entities in the object, the pronoun is resolved by the named entity in subject, i.e. Kamal Rao. The 'he' after the quotes text is in the subject. This 'he' is replaced with the name entity in subject of the previous sentence as there no appropriate named entities in the subject of the sentence containing the pronoun. Also, the speaker of the previous sentence is used to replace the pronoun appearing in the subject of the third sentence. For resolution of the pronouns I, Me and My, the algorithm is as represented by the flowchart in figure 5. Considering the previous example, in the first sentence I resolves to Kamal Rao. Also, I in the next sentence refers to he which is resolved to Kamal Rao.

For We, the system identifies if the pronoun occurs in a quoted

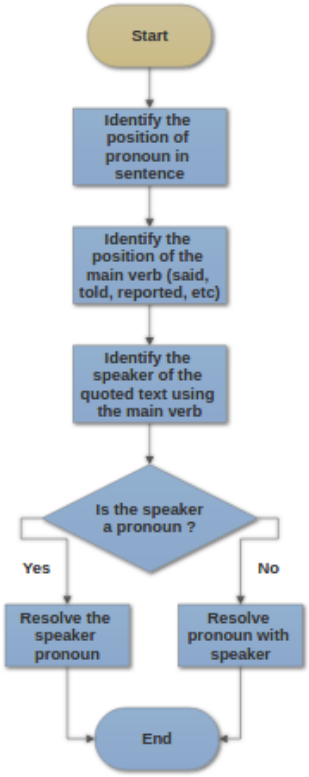

Figure 5: Resolution of pronouns of the class I (I, Me, My)

sentence. If the sentence is a quoted sentence then the pronoun is resolved to the speaker of the quote. The next step is to identify chunks of the type plural and animate. The pronoun 'We' is largely associated with the problem of split antecedents where if the sentence contains two named entities of the same type i.e. their properties are the same, the pronoun is resolved to these named entities in conjunction. 'We' may be used in the case where a speaker associates themselves with an organization and collectively refers the two with the pronoun.
Pronouns of the class they (they, them, their) are complicated as they can be used for animate or inanimate, singular (gender independent) or plural entities. The system currently identifies a general case for these pronouns. The approach is to check whether the word

preceding the pronoun to be resolved is a verb, the flag is set. A set flag indicates that the pronoun has to replaced with an animate chunk. All the chunks before the pronoun are checked, if the flag is set, the latest animate plural chunk is returned, else the latest plural chunk (animate or inanimate) is returned.
e.g. In all, 23 persons, many of them former corporators, MLAs and some even sitting corporators or seeking tickets for their wives, joined the Youth Party or Samaj Sena in the recent past.
In the example them refers to 23 persons their refers to corporators.
A generalized form of 'you' in a text indicates that the speaker is addressing the reader. The system does not consider such occurrences. For other cases the position of pronoun you is observed. If it is inside quotes then it needs to be resolved, else pronoun you refers to the all readers, so appropriately the flag is set. If the flag is set, then according to the position of pronoun you and words like said, told or added, a rule is fired and a chunk is returned.
e.g. There is a war going on between Youth Party and Congregation Party. "Mohan ji, you are not working for the benefit of the country," said Ram. Ram told media addressing to Mohan " Mohan you are destroying the nation". In the first sentence you resolves to Mohan ji. In the first sentence you resolves to Mohan.

3.5 Anaphora Resolution

3.5.1 Resolve Abbreviation. Many times in texts, a named entity is expressed in its abbreviation form which makes resolution of abbreviations an important case for anaphora resolution. To co-relate abbreviations and their expansions we use two approaches. The first approach is a static approach where the system maintains a standard list of abbreviations stored in the ontology with corresponding expansions. The tool identifies an abbreviation in the text and queries the ontology to extract the associated expansion. For newly occurring abbreviations, the words that occur previously are examined, if these words match the acronym under question the pair is identified and stored in a map. Expansion of abbreviations helps give more context for identification of named entities represented in their shorter forms, using the Stanford NER.

3.5.2 Alias Resolution. A named entity may be referred by different parts of the same name. In case of a 'PERSON', the first or last name, in case of 'LOCATION' and 'ORGANIZATION, an abbreviation or an alias given to it (requires world knowledge). The system identifies abbreviations using the previous mentioned approach. For cases of a part of the complete name being mentioned to refer to a previously mentioned entity the system proposes the following approach. It maintains a list that contains the first occurrences of named entities in the text, a unique entity list. Each named entity has a list of aliases that are identified. On encountering an entity mention, the system checks its occurrence in the unique entity list. It then checks whether an alias has been identified by the same name. The final case to check for similarity between the main entity and the current entity. If the match is satisfactory the current entity is added to the main entity's alias list. In the example mentioned in section 3.4, 'Tiwari' resolves to 'Vijay Tiwari' and 'Dal' resolves to 'Jan Sena Dal'.

3.5.3 Add Designations to the Named Entities. Designations are a type of alias that can be used for a 'PERSON'. Thus the system explores identification of designations for a named entity and resolution of their occurrences in the text. For each named entity, the location of the first occurrence of the name entity is identified. A noun phrase that can serve as a designation is identified. If the previous case fails, then the noun phrase after the named entity can be searched. Prepositions, punctuation marks and determiners, if present between the nouns can be ignored. The system maintains a designation map with the designation string as a key and the associated named entity as its value. The occurrence of the designation strings is checked, if present it is deemed as a named entity and is added to the alias list of the named entity that the designation belongs to. The designation map is used in this case. Subparts of the associated designation string may also be used for referring a named entity, candidates can be generated for every designation and then resolved. Considering the example mentioned in section 3.4, designations 'party leader' will be identified for 'Vijay Tiwari'. Occurrence of 'party leader' in the next sentence is identified as an alias for 'Vijay Tiwari'.

3.6 Relationship Identification

Firstly all the entities are identified, not only named entities but pronouns, designations, aliases, chunks. An annotation is created for each entity that connects it to a tag that defines its category. Relationships between different entities may be classified into two types, resolves and alias. These represent the output obtained from the previous co-reference resolving modules. Each relationship is identified between two entity arguments. Figure 6 shows a part of the automatically generated annotation file. Each entity tag starts with the letter 'T' and each relationship tag with the letter 'R'. The second column gives the tag associated. For entities the last two columns maintain the start and end character indexes of the text and for relationships they give two entity arguments.

```
T1  Location 0 6
T2  Organization 402 424
T3  Organization 477 499
R1  alias Arg1:T3 Arg2:T2
T4  Organization 549 571
R2  alias Arg1:T4 Arg2:T2
T5  Organization 662 684
R3  alias Arg1:T5 Arg2:T2
T6  Organization 857 879
R4  alias Arg1:T6 Arg2:T2
T7  Organization 986 1008
R5  alias Arg1:T7 Arg2:T2
T8  Organization 1191 1213
R6  alias Arg1:T8 Arg2:T2
T9  Organization 1266 1288
```

Figure 6: The annotation file

This annotation file is then visualized by using the BRAT - Rapid Annotation Tool. A subset of sentences visualized using BRAT are shown in figure 7.

4 RESULTS AND DISCUSSIONS

The performance measure of the proposed system has been captured in Table 1: Evaluation parameters for the system's performance

Table 1

Module	Precision	Recall	F-measure
Pronoun Resolution	0.6366	0.7156	0.6431
Anaphora Resolution	0.7581	0.6793	0.7165
Chunk Extraction	0.8934	0.8853	0.8893

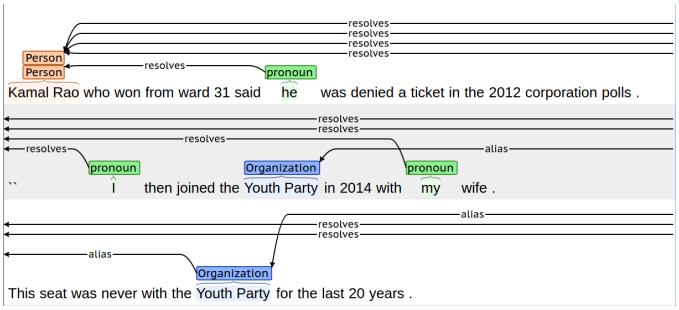

Figure 7: Visualization using BRAT

are precision, recall and f-measure. Precision is the fraction of instances correctly resolved out of the total instances resolved. Recall is the fraction of instances that are resolved correctly out of the total instances that should have been resolved. F-measure is the the harmonic mean of precision and recall. The discourse linking tool was tested on the corpus mentioned in section 3.1. The preliminary results for all the three modules are promising.

With the proposed methodology we encountered certain challenges. Every relationship required a set of parameters for its identification, creation of a feature set that would hold all the required information was a challenging task. The next challenge observed was, incorporation of a deterministic rule set for pronoun and anaphora resolution. It involved construction of generalized rules while keeping in mind the existence of specific natured instances that have not yet been realized. Thus the first step of improvement in this case can be done by identifying different rules and mapping a hierarchical relationship between them.

Another problem that directly affects the performance of the existing system is that of error propagation due to its pipeline architecture. For example, if the part of speech (POS) for a word is tagged wrongly it may result into formation of an incorrect chunk as the chunking module heavily relies on the POS tags. Incorrect chunks may further result in an erroneous result in the resolution module and thus the final results are affected largely by this problem of error propagation.

5 SIGNIFICANCE OF THE SYSTEM

The system represents the discourse in the form of relationships in an annotation file. This annotation file can be used to train the NameFinder module in the OpenNLP project of Apache Group. OpenNLP gives native support for BRAT annotation files. Furthermore, these annotation files can easily be converted to xml files. XML provides a language and technology independent method of storing data. These relationships can also be visualized in a graph database. With such a realization of this tool, a Question-Answer (QA) system and automatic summary generating software can be created. The named entities and verbs can form nodes in the graph. Traversing the incoming and outgoing edges of a node will help in answering questions about a certain named entity or a verb. Also, depending on the number of connections of a node, the one with the highest number of edges associated with it can be considered as the main topic of the text. Using this knowledge, a summary can also be generated from the text.

6 CONCLUSIONS

After incorporating the domain knowledge with co-referencing we are able to get a linked discourse for the given text. Although for visualization we have used BRAT-Rapid Annotation tool, the discourse can be also be stored in the form of a graph database. The generated annotations provide tokens and their links, hence, logically creating a discourse linked sequence. This automated English Discourse linking application can potentially have many usages in NLG related developments and other applications of NLP including QA systems, text summarization etc.

REFERENCES

[1] Rashmi Prasad, Bonnie Webber, Alan Lee, Sameer Pradhan and Aravind Joshi, *Bridging Sentential and Discourse-level Semantics through Clausal Adjuncts*, Proceedings of the EMNLP 2015 Workshop on Linking Models of Lexical, Sentential and Discourse-level Semantics, pages 64âĂŞ69. Lisboa, Portugal, 2015.

[2] Heeyoung Lee, Yves Peirsman, Angel Chang, Nathanael Chambers, Mihai Surdeanu and Dan Jurafsky, *Stanford's Multi-Pass Sieve Coreference Resolution System at the CoNLL-2011 Shared Task*, Proceedings of the Fifteenth Conference on Computational Natural Language Learning. Stanford NLP Group Stanford University, Stanford, CA 94305, 2011.

[3] Kevin Clark and Christopher Manning, *Deep Reinforcement Learning for Mention-Ranking Coreference Models*, EMNLP 2016 Computer Science Department, Stanford University, 2016.

[4] Mark Sammons, Christos Christodoulopoulos, Parisa Kordjamshidi, Daniel Khashabi,Vivek Srikumar, Paul Vijayakumar, Mazin Bokhari, Xinbo Wu and Dan Roth, *Feature Extraction for NLP, Simplified.* Department of Computer Science,University of Illinois, Urbana-Champaign.

[5] Leon Derczynski, Diana Maynard, Giuseppe Rizzo , Marieke van Erp,Genevieve Gorrell , Raphael Troncy , Johann Petrak and Kalina Bontcheva, *Analysis of named entity recognition and linking for tweets.* Information Processing & Management Volume 51, Issue 2, 2015.

[6] Samet Atdag and Vincent Labatut, *A Comparison of Named Entity Recognition Tools Applied to Biographical Texts.* Harlow, England: Addison-Wesley, 1999.