Asian Journal of Information Technology 15 (16): 2831-2838, 2016
ISSN: 1682-3915
© Medwell Journals, 2016

Two Layer Machine Learning Approach for Mining Referential Entities for a Morphologically Rich Language

Vijay Sundar Ram and Sobha Lalitha Devi
AU-KBC Research Centre, Anna University, MIT Campus, Chennai, India

Abstract: Business Intelligence (BI), a technology-driven process and presenting actionable information has become important for improvement of organisation, business units etc. BI requires mining information from a huge volume of unstructured text data. This mining task requires sophisticated natural language processing tasks. One of the crucial tasks is identifying the chain of referential entities in the given text which is described as coreference resolution. Coreference is the referent in one expression of the same referent in another expression and the referents must exist in the real world. Coreference chain is formed by connecting entities referring to same entity. We approach this resolution task for a morphologically rich language, Tamil as two subtasks and use two machine learning approaches. The two subtasks, the pronominal resolution and noun phrase coreferencing is done using Tree Conditional Random Fields (Tree CRFs) and Support Vector Machines (SVM), respectively. Coreference chains are evaluated with standard metrics and the results are encouraging.

Key words: Chain of referential entities, coreference resolution, pronominal resolution, noun phrase coreferencing, Tree CRFs, support vector machines, Tamil, morphologically rich language

INTRODUCTION

Business Intelligence (BI) includes a variety of tools and methodologies to collect and analyse data from various source and create a report on the data. BI plays a vital role in driving person, organisation, business units, etc. BI analyse information mined from various data sources. One of the major challenges in data mining is extracting information from the unstructured text data which is in huge volume and exponentially grows day by day. Processing of the unstructured text requires sophisticated natural language processing tools which work in syntactic, syntactico-semantic and semantic level. One of the important tools required for text processing is the referential entity chain identifier to identify the referential entities in the given text. This task of identifying the referential entity chain is called coreference resolution. In this study, we describe an approach using two machine leaning techniques for analysing a morphologically rich language, Tamil. Coreference resolution is the task of identifying the real world entities mentioned in the documents. Various mentions or Noun Phrases (NPs) referring to the same entities are connected to form a coreference chain. Coreference holds between two entities which can be definite noun phrases, demonstrative noun phrases, proper names, appositives and pronouns. Coreference resolution is required in various natural language applications such as information extraction, question answering system, profile building system, information retrieval, machine translation, etc. In this study, we describe coreference in Tamil text and an approach to coreference resolution in Tamil text. Though, coreference resolution is well attempted research area in various European languages, it is not attempted in Indian languages. In Indian languages automatic resolution of anaphoric pronouns is developed in languages such as Hindi, Bengali, Tamil and Punjabi. The present work is first of its kind. A survey on automatic coreference resolution engines in various languages are presented as follows.

The research in building automatic resolution of anaphoric entities such as pronouns started in late 70 and early 80s with Hobbs algorithms as one of the earliest approach. In early years of 2000 there was published work in coreference resolution task. Soon *et al*. (2001) has presented a work on noun phrase coreference resolution using decision tree approach in which he had used 12 features that were learned from the corpus. Cardie and Wagstaff (1999) considered noun phrase coreference resolution as a clustering task. They came up with an unsupervised algorithm, flexible for co-ordinating the application of context-independent and context dependent constraints and preferences for accurately partitioning the noun phrase into coreference clusters. Ng and Cardie (2002) has investigated on the anaphoricity

Corresponding Author: Vijay Sundar Ram, AU-KBC Research Centre, Anna University, MIT Campus, Chennai, India of noun phrase and the approaches to identify the anaphoric and non-anaphoric noun phrases. Versley has used maximum entropy model to find weights for the hard and soft constraints in finding noun phrase coreference resolution in German newspapers. Haghighi and Klein (2009) presented a deterministic system which is entirely driven by syntactic and semantic compatibility as learned from a large unlabelled corpus. Stoyanov *et al.* (2010) has presented the various levels of challenges in noun phrase coreference. They have discussed issues from named entity task to coreference revolver. Ng (2010) has presented a survey report on noun phrase coreference resolution. He has broadly classified the works as mention-pair model, entity-mention model and ranking model. He has presented the merit and disadvantages in finer level. He has also discussed about the knowledge sources used and the evolution metrics used. CoNLL Shared task titled “Modeling Unrestricted Coreference in Ontonotes” boosted the coreference resolution task in English (Pradhan *et al.*, 2011). Ram and Devi (2012, 2013) presented a coreference resolution system for English using tree conditional random fields for resolving pronouns and linear conditional random fields for resolving noun phrase coreference resolution. Stoyanov *et al.* (2010) have proposed an approach where the easy coreference pairs are addressed first followed by harder pairs (which require semantics and world knowledge). Durrett and Klein (2014) has also presented similar approach of getting easy victories and uphill battles in coreference resolution, using automatic capture of heuristics with a number of homogeneous feature templates examining the shallow properties of the mentions. Martschat (2013) presented an unsupervised model for coreference resolution that casts the problem as a clustering task in a directed labeled weighted multigraph. Yang *et al.* (2015) has designed a feature based classifier and an embedding based ranker which were tailored to model mention reference relation for proper names and common nouns. Clark and Manning (1999) used entity-level information in building coreference chain incrementally.

In this study, we present an approach for coreference resolution in Tamil text using machine learning techniques. We have approached this as three sub tasks and it is described in detail in the further study.

MATERIALS AND METHODS

Coreference in Tamil: Coreference resolution is the task of identifying the real world entities mentioned in the text. All the mentions that refer to the same entity are grouped together to form a coreference chain for that mention. We try to analyse the coreference in Tamil text. Tamil, a South Dravidian language is morphologically rich and highly agglutinative. Noun and verbs in Tamil are suffixed with multiple suffixes. Nouns are suffixed with plural and case markers, verbs are suffixed with tense, aspect, modal and person, number gender suffixes. Conditional, relative participle markers are also suffixed to the verb. The suffixes carry rich morphological information. Tamil is relative free word order language has rigid structures within the clauses. The subject has person number gender agreement with the finite verb. There is gender and number distinction in most of the pronouns. First person pronouns and second person pronouns do not have gender distinction whereas number distinction such as ‘nii’ (you), ‘naan’ (I), ‘niingkal’ (you-plural) exists. Third person pronouns have number and gender distinction such as ‘avan’ (he), ‘aval’ (she), ‘atu’ (it). In third person pronoun, plural pronoun ‘avarkal’ refers to both masculine and feminine genders and also represents honorific. Similar to other languages the mentions (real world entities) can occur as a noun phrase and pronouns. The entities can be referred with complete NP, partial NP, pronouns, relation words, its acronym, definite description, etc. The coreference chains in Tamil are explained further with the examples below.

Example 1:

- Aang saan suu ki miyaanmaaril, Aung San Suu Kyi Myanmar(N)+loc, piRanthavar. was_born (Aung Sa Suu Kyi was born in Myanmar)
- Avar annaattu raaNuvaththaal 15, She(PN) that+country(N) military(N)+ins 15, aaNtukkaalam viittukkaavalil, years_periord(N) house_arrest(N)+loc ataikkappattaar, put(V), (She was put under house arrest for 15 year)
- Suu kikku amaithikkaaNa noopal, Suu Kyi(N)+dat peace(N)+adv Nobel(N), paricu valangkappattathu. price(N) give(V) , (Nobel prize for peace was given to Suu Kyi)

In example 1, there are three sentences and it has two real world entities ‘aang saan suukki’ (Aung San Suu Kyi) and ‘miyaanmaar’ (Myanmar). These two real world entities has occurred across these three sentences. On the identifying, the entities and connecting the entities forms the following two coreference chains in Fig. 1 and 2.

Here, the real world entity “aang saan suukki” is referred with pronoun ‘avar’ in the second sentence and partial noun-phrase ‘suukikku’ in the third sentence.

Here in the above coreference chain, real word entity ‘miyaanmaar’ which has occurred in the first sentence is referred by appositive name ‘naatu’ in the second

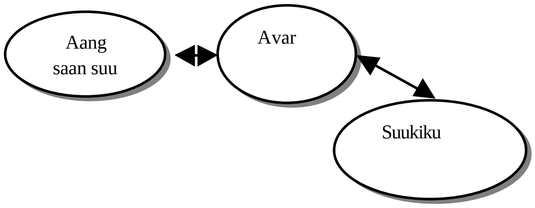

Fig. 1: Coreference chain for the entity 'aang saan su ki'

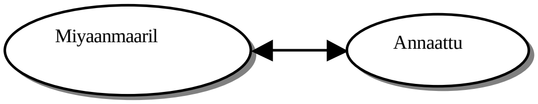

Fig. 2: Coreference chain for the entity 'miyaannaar'

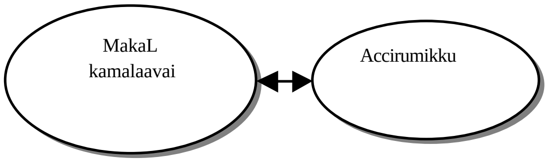

Fig. 3: Coreference chain for the entity 'makaL kamalaa'

sentence. Here, the word 'naatu' is prefixed with a demonstrator and suffixed with a dative marker.

Example 2:

- Raamu than makaL, Ramu(N) his(PN) daughter(N), kamalaavai viilaavukku, Kamala(N)+acc function(N)+dat, alaiththuvanthaar. brought(V) (Ramu brought his daughter Kamal to the function.)
- Accirumikku inippukaL pidiththathu. That girl(N)+dat sweets(N) like(V) (That girl liked sweets)

In example 2, there are two entities, 'raamu' (Ramu) and 'makaL kamalaa' (Daughter Kamala). One of the entities has occurred twice. The coreference chain that exists in example 2 is as Fig. 3.

Here the noun phrase 'makaL kamalaavai' which is suffixed with accusative marker is referred by an appositive entity 'accirumikku' which is suffixed with a dative marker.

Example 3:

- neRRu pirathamar narenthira moodi, Yesterday prime minister Narandra Modi, metrovil payanam, ceythaar. Metro(N)+loc journey(N) do(V)+past+3SH (Yesterday Prime Minister Narendra Modi travelled by metro)
- Pirathamaraik kaNtu makkaL, Prime Minister (N)+Acc see(V) people(N), makilnthanar. feel_happy (V)+past+3PL (People were happy seeing the prime minister)

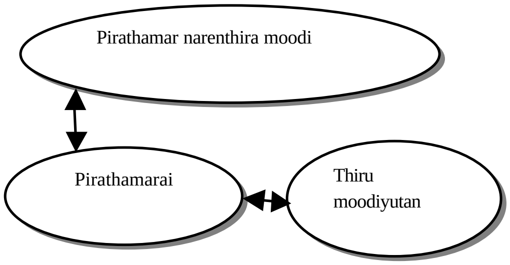

Fig. 4: Coreference chain for the entity 'pirathamar narenthira moodi'

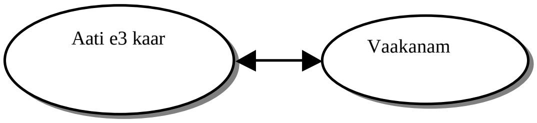

Fig. 5: Coreference chain for the entity 'aati e3 kaar'

- Thiru moodiyutan makkaL, Mr Modi(N)+gen people(N) Pukaipatam eetuththukkoNtanar. photograph(N) take(V)+past+3PL, (Mr Modi took photos with people)

In example 3, the entity 'Pirathamar Narenthira Moodi' (Prime Minister Narandra Modi) has occurred in other two sentences using definite description and partial noun. The coreference chain that exists in example 3 is as Fig. 4.

Here, the real world entity 'pirathamar narenthira moodi' is referred by two noun phrase, one a definite description 'pirathamar' and partial noun phrase 'thiru moodi'.

Example 4:

- En naNpan oru aati e3 kaar, My friend(N) one Audi_e3 car vaangkinaar. buy(V)+past+3SH (My friend bought Audi e3 car)
- Antha vaakanam mikavum, that vehicle (N) very (ADJ), vacathiyaaka irukkirathu. comfort (N)+ADV present (V) (that vehicle is very comfortable)

The coreference chain present in example 4 is the entity 'aati e3 kaar' and its appositive relation 'vaakanam''. The coreference chain is as Fig. 5.

Here the real world entity 'aati e3kaar' is referred by its appositive 'vaakanam'. The identification of appositive relation requires world-knowledge to resolve it.

Our approach: In the coreference resolution task, most of the published works try to address the identification of anaphoric pronouns and anaphoric noun phrases

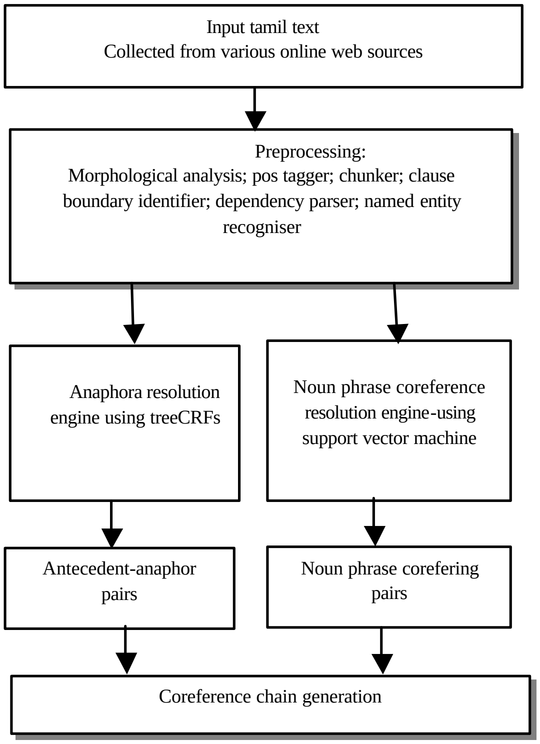
Input tamil text
Collected from various online web sources
Preprocessing:
Morphological analysis; pos tagger; chunker; clause boundary identifier; dependency parser; named entity recogniser
Anaphora resolution engine using treeCRFs
Noun phrase coreference resolution engine-using support vector machine
Antecedent-anaphor pairs
Noun phrase corefering pairs
Coreference chain generation

Fig. 6: Flow of the Tamil coreference resolution engine

together as a single process. Noun phrase mentions such as complete string match noun phrase, partial string match noun phrase, appositive for the entities do not have a strong syntactic relation compared to anaphoric pronouns and its referent noun phrases. The resolution of the anaphoric pronouns requires more syntactic cues. Since, Tamil is morphological rich languages and suffixes carries lot of information we require pronominal resolution to be performed separately.

Our three tasks are as follows noun-phrase coreference resolution, pronominal resolution and chain formation. The architecture of the coreference resolution engine is presented in Fig. 6. In the first two sub tasks we pair the entities which refer to the same entity. In the noun-phrase coreference, we try to resolve the pair of entities which has relation such as complete string match, partial string match, appositive and acronym. The anaphoric pronouns are resolved and paired with its antecedent in the pronominal resolution task. Once the pronominal resolution and noun phrase coreference resolution is performed, using the paired entities the coreference chain is built.

Pronominal resolution: Pronominal resolution is the task of identifying the antecedent (referent) of the pronouns. The antecedent occurs in the text before the pronoun. Here, we use machine learning technique, Tree Conditional Random Fields (TreeCRFs) for this task. The approach to this is inspired from (Ram *et al*., 2013). In Tree CRFs, general graphs are used instead of linear chains in linear CRFs. TreeCRFs does a joint prediction of multiple outputs. So, we try to identify the anaphor and its corresponding antecedent using the Tree CRFs. It uses a Tree based Reparameterization (TRP) algorithm which includes Belief Propagation (BP), approximate inference algorithm for parameter learning task. Tamil being relatively free word order language requires the machine learning technique to learn the structure of the sentences. Tree CRFs is suitable for this task as in Tree CRFs exact computations over spanning trees of the full graph is performed. This helps in learning long distance relation and makes this suitable for this task. The Tree CRFs forms a clique between two nodes and tries to predict the labels of the two nodes based on the features between the two nodes. The detailed description of Tree CRFs is available by Sutton and McCallum (2006). Here, we have used an open source version of Tree CRFs by Sutton.

Features identified for tree CRFs: The input sentences are shallow parsed with morphanalyser, Part-of-Speech (POS) tagger, chunker and clause boundary identifier and parsed with dependency parser.

We have used inhouse developed dependency parser based on Paninian dependency scheme. Chunk (noun phrase, verb phrase) is considered as the basic unit. The scheme is based on the modifier-modified relationship. The relations are of two types: karakas and others. The direct syntactic relations between the noun and verb are captured by karaka relation (k1, k2, k3, k4, k5, r6, k7) and relational concept of models (adjuncts) are captured by other relations. Rafiya *et al*. (2008) has presented a detailed description of the scheme. The karaka relations used in this tagging is presented in Table 1. From the shallow parsed text the following features are extracted:

- Node features
- Adjacent features
- Edge-clique features

Node features: Node features denote the independent features of the nodes both anaphor and the possible antecedent node. The features are as follows:

- Its syntactic category, i.e., subject, object, indirect object
- Its dependency relation
- Type of the pronoun
- Case markers suffixed to it
- Its position with respect to clausal boundaries

Table 1: Karaka relations

Relation	Description
k1	Doer/agent/subject
k2	Object/patient
k3	Instrument
k4	Recipient
k5	Source
r6	Genitive/possessive
k7p	location in space
k7t	location in time

Adjacent features: These features are obtained with respect to its surrounding nodes such as information regarding its parent node, type of the verb the nodes is attached.

Edge-clique feature: Edge-Clique features include features related to anaphor node and the possible antecedent node. These features are as follows:

- Both the nodes are having same parent node
- Both nodes are pronouns

The training for this task is done by collecting anaphor and antecedent pairs as the positive set and the anaphor and the noun phrase which matched with person number gender of the pronoun in the above 4 sentences and the current sentence are collected as negative set. The above mentioned features are extracted for the anaphor and its antecedents and the PNG matched noun phrases. These set of data after feature extraction is presented to the machine learning technique (tree CRFs) to form a language model. In the testing phase, for each pronoun, the noun phrases which occur in the current sentence where the pronoun has occurred and four sentences above from the current sentence are collected. In those NPs which match in person, number and gender with the pronoun are chosen as possible candidate NPs. These possible candidate NPs are paired with the pronoun and the features are extracted for each pair. The features extracted from each pair are presented to the machine learning technique along with the language model for resolution.

Noun phrase coreference resolution: In noun phrase conferencing, we try to identify the pair of entities which have complete string match relation, partial string match relation, appositive relation, definite description and acronym. We have used Support Vector Machine (SVM), a Machine Learning (ML) technique which is popular for classification task. Support vector machines are supervised learning methods used for classification and regression analysis. They are used for two group classification problems. SVMs map the input vector non-linearly to very high dimension feature space. In this space a linear decision surface is constructed. The special properties of linear decision surface ensure high generalization ability of the learning machine. The small amount of training data which determines the margin to construct optimal plane is called support vectors. Once the SVMs is trained, we need to simply determine on which side of the decision boundary a given test pattern lies and assign the corresponding class label . In this research, we have used YamCha tool. YamCha is a generic, customizable and open source text chunker directed toward a lot of NLP tasks, such as PoS tagging, named entity recognition, base NP chunking and text Chunking. It is using a state-of-the-art ML algorithm SVMs (Kudo and Matsumoto, 2000). Here in this task, the features of the two entities are examined and the ML technique should give a binary output. The identification of features is the important task while using the ML technique. The features used for this task are explained in the following study.

Features used in SVM: We have come up with general features and specific features. The detailed description of the features is presented as below. Features used in SVM from NP_i and Np_j. General features and positional feature are as.

Sentence initial position: 1 If the NP_i occurs in the starting of the sentence else 0; 1 if the NP_j occurs in the starting of the sentence else 0.

After PP: 1 if the NP_i occurs after a preposition else 0; 1 if the NP_j occurs after a preposition else 0.

End of the sentence: 1 if the NP_i occurs in the ending of the sentence else 0; 1 if the NP_j occurs in the ending of the sentence else 0.

NP type

Definite NP: 1 if the NP_i starts with ‘the’ else 0; 1 if the NP_j starts with ‘the’ else 0.

Demonstrative NPL: 1 if the NP_i starts with demonstratives such as ‘that’, ‘this’, ‘these’ else 0; 1 if the NP_j starts with demonstratives ‘the’ else 0. Proper name: 1 if NP_i is a proper noun, else 0 ; 1 if NP_j is a proper noun, else 0.

Specific features

Complete match and full string match: 1 if NP_i and NP_j has full string match, else 0. Alias: 1 if NP_i is an Alias of NP_j or vice-versa, else 0. Appositive: 1 if NP_i , NP_j has appositive relation, else 0. Partial match.

Percentage of match: Percentage of string match between NP$_i$ and Np$_j$. Distance number of lines between NP$_i$ and NP$_j$.

Capital letter: 1 if NP$_i$ is in capital letter, else 0; 1 if NP$_j$ is in capital letter, else 0. Relation words, head noun head Noun in NP$_i$ and Np$_j$. Distance number of lines between NP$_i$ and NP$_j$

Similar to the pronominal resolution task using tree CRFs, noun phrase coreference resolution engine using SVM requires training and testing. The training data is prepared by collecting the pair of corefering entities from the pre-processed and annotated data and above mentioned features are extracted for these entity pairs. These pairs form the positive pairs. The negative pair of entities are formed by collecting two non-corefering NPs. For these entities the above mentioned features are extracted. The collection of the positive and negative pairs of entities forms the training data. The training data is given to the SVM, to generate the language model. The testing data is first pre-processed and the features for the pairs of entities are collected and these pairs of entities are given to the classification engine with the language model which was generated in the training phase.

Coreference chain generation: The coreference chain is built using the output obtained from the anaphora resolution and noun phrase coreference engines. The anaphora resolution engine gives the anaphor-antecedent pair and the noun phrase coreference resolution engine gives pair of entities which refer with relations such as complete string match, partial string match, appositives and acronym. Using the obtained pairs of entities the coreference chain is built.

RESULTS AND DISCUSSION

Corpus was collected from tourism web pages and anaphor-antecedents and noun phrase coreference resolution entities are manually annotated. The annotated corpus is enriched with shallow parsing tool such as morphological analyser, part-of-speech tagger, chunker, clause boundary identifier, dependency parser and named entity recogniser. The data is presented in column format where the column information are as follows:

- 1st column has the document id
- 2nd column has the sentence id
- 3rd column has the word id

Followed by word, its POS tag, chunk information, NE information, dependency tag and the coreference information. Statistics of the tagged corpus is presented in Table 2.

Table 2: Corpus statistics

Variable	Training data	Testing data
Anaphor-antecedent	925	609
Noun phrase corefering pairs	565	378

Table 3: Performance of the coreference engine

Variables	Precision (%)	Recall (%)	F-measure (%)
MUC	49.56	60.72	54.57
B^3	61.32	77.89	68.61
CEAFe	53.72	42.59	47.51

Table 4: Performance of the anaphora resolution engine

Variable	Precision (%)	Recall (%)	F-measure (%)
Anaphora resolution	73.83	70.23	71.98

The annotated data is divided into 80/20 portions, where the 80% portion of the corpus was used as training data and 20% portion of the corpus was used as testing data. The language models are built for the pronominal resolution using tree CRFs technique and noun phrase resolution model using SVM techniques. The testing data is given to the anaphora resolution engine. After resolving the anaphoric pronouns, the output is given to the noun phrase resolution engine and the combined output is given to the coreference chain generation module.

The performance evaluation of anaphora resolution is generally using metrics such as precision, recall and F-measure. In coreference chain, we try to measure based on the anaphoric links identified and the correctness of the anaphor and antecedent. The different methods used for evaluating the coreference chains are link-based method, set-based method and alignment-based method. In link-based approach, the precision is ratio of number of correctly identified anaphoric links to the total number of anaphoric links the system has resolved. And recall is the ratio of number of correctly identified anaphoric links to the anaphoric links in the gold annotated corpus. In set-based methodology, the portion of the output are compared with gold standard output. The alignment-based method is common to the previous two methods and induced the idea of alignment between the gold and the system partitions. The evaluation of the coreference chain is done using the standard metric such as MUC, B^3 and CEAFe (Bagga and Baldwin, 1998; Recasens and Hovy, 2011; Vilain *et al*., 1995). The performance scores for the coreference chain is presented in Table 3. The performance scores for pronominal resolution are presented in Table 4.

On analysing the coreference chains the entities with complete string match are perfectly identified. Entities with partial string match have more false positives. Two NPs which have partial match between them, can possibly be a corefering noun phrase, but it is not true in all case. There are a large number of non corefering NP which have partial match between them. And partial matching NPs may refer two different entities, example "thitta kulu" (planning commission) and "cuya uthavi kulu" (self help groups). Here both the noun phrases has 'kulu' as the common partial match word. But both the NPs are completely different. These bring more number of false positives. The challenge lies more in identification of appositive relations. Named entity tags helps in identifying the relation such as 'inthiya' (India) and 'annaatu' (that country). By using resources such as wordnet, thesaurus, we can handle the NP pairs which does not have string match. More syntactic features should be used in identifying correct partial match pairs and disambiguating the non-corefering pairs. Performance on anaphora resolution has to be improved. On analysing the anaphora resolution output, the resolution of third person neuter pronoun 'atu' has to be improved. The pronoun 'atu' generally has more number of possible antecedents and its resolution is poor. Consider the following example 5.

Example 5:

- Thanjai periya kovil koopuram, Thanjai (N) big (ADJ) temple (N) tower (N) ore paaraiyaal, kattapattathu. one rock(N)+INS built(V)+past+3sn (Thanjur big temple tower was built with one rock)
- Atu yaanaikalaal meele It (PN) elephants (N)+INS top (N), koNtuc_cellappattathu. take(V)+past+3sn (it was taken to the top by elephants)

In the above example 5, there are two sentences, the second sentence has third person neuter pronoun 'atu'. The possible candidate for this pronoun is 'thanjai periya kovil', 'koopuram', 'ore paaraiyaal'. In the first sentence, there is a genitive drop. The subject noun phrase should have occurred with the genitive marker as follows, 'thanjai periya kovilyin koopuram' (Thanjai periya kovil's Kopuram). The genitive marker 'in' suffixed to the noun 'kovil' is dropped. The anaphora resolution engine chooses the 'thanjai periya kovil' which has occurred in the subject position. Here, the correct antecedent for the anaphor 'atu' in the second sentence is 'ore paaraiyaal' which has instrumental case suffixed with it and it is not in the subject position. Resolution of honorific pronouns is also poor as these pronouns can have more number of possible antecedents with both the genders.

CONCLUSION

We present Tamil coreference resolution engine which uses Tree CRFs and SVM which is a crucial tool in analysing unstructured data. We have divided the resolution task into pronominal resolution and noun phrase coreference resolution where we have used tree CRFs for pronominal resolution and SVM for noun phrase coreference resolution. Corefering entities with complete string match are identified perfectly and it is an easy task. Entities with partial string match introduce false positives. These partial string match entities require validation while choosing the entities. Identification of entities with appositive relation is tougher. In the present task, we have used the named entity tags for identifying the appositive relation. Though, we have used these NE features for appositive identification, the resolution of entities with appositive relation is poor. We require knowledge-sources such as wordnet and thesaurus for further identification. Resolution of anaphoric pronouns plays an important role and resolution of the third person pronoun and honorific pronouns has to be improved. Coreference resolution engine can used to build automatic profile of a person, organisation, etc., which is required for a Business intelligence.

REFERENCES

Bagga, A. and B. Baldwin, 1998. Algorithms for scoring coreference chains. Proceedings of the First International Conference on Language Resources and Evaluation Workshop on Linguistics, May 28-30, 1998, Duke University, Durham, North Carolina, pp: 563-566.

Cardie, C. and K. Wagstaff, 1999. Noun phrase coreference as clustering. Proceedings of the Joint Sigdat Conference on Empirical Methods in Natural Language Processing and Very Large Corpora, June 21-22, 1999, Cornell University, Ithaca, New York, USA., pp: 82-89.

Durrett, G. and D. Klein, 2014. A joint model for entity analysis: Coreference, typing and linking. Trans. Assoc. Comput. Ling., 2: 477-490.

Haghighi, A. and D. Klein, 2009. Simple coreference resolution with rich syntactic and semantic features. roceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, August 6-7, 2009, Association for Computational Linguistics, Stroudsburg, Pennsylvania, USA, pp: 1152-1161.

Kudoh, T. and Y. Matsumoto, 2000. Use of support vector learning for chunk identification. Proceedings of the 2nd Workshop on Learning Language in Logic and the 4th Conference on Computational Natural Language Learning, September 13-14, 2000, Association for Computational Linguistics, Stroudsburg, Pennsylvania, USA, pp: 142-144.

Martschat, S., 2013. Multigraph clustering for unsupervised coreference resolution. Proceedings of the Workshop on ACL Student Research, August 4-9, 2013, ACL, Sofia, Bulgaria, pp: 81-88.

Ng, V. and C. Cardie, 2002. Improving machine learning approaches to coreference resolution. Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, July 6-12, 2002, Association for Computational Linguistics, Philadelphia, Pennsylvania, USA, pp: 104-111.

Ng, V., 2010. Supervised noun phrase coreference research: The first fifteen years. Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, July 11-16, 2010, Association for Computational Linguistics, Stroudsburg, Pennsylvania, USA, pp: 1396-1411.

Pradhan, S., L. Ramshaw, M. Marcus, M. Palmer and R. Weischedel *et al*., 2011. Conll-2011 shared task: Modeling unrestricted coreference in ontonotes. Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task, June 23-24, 2011, Association for Computational Linguistics, Stroudsburg, Pennsylvania, USA, ISBN:9781937284084, pp: 1-27.

Rafiya, B., H. Samar, D. Arun, M.S. Dipti and B. Lakshmi *et al*., 2008. Dependency annotation scheme for indian languages. International Institute of Information Technology, Hyderabad, India,

Ram, R.V.S. and S.L. Devi, 2012. Coreference Resolution Using Tree CRFs. In: Intelligent Text Processing and Computational Linguistics. Alexander, G. (Ed.). Springer, Berlin, Germany, ISBN:978-3-642-28603-2, pp: 285.

Ram, R.V.S. and S.L. Devi, 2013. Pronominal resolution in Tamil using tree CRFs. Proceedings of the 2013 International Conference on Asian Language Processing (IALP), August 17-19, 2013, IEEE, Chennai,India,ISBN:978-0-7695-5063-3,pp:197-200.

Recasens, M. and E. Hovy, 2011. BLANC: Implementing the rand index for coreference evaluation. Nat. Lang. Eng., 17: 485-510.

Soon, W.M., H.T. Ng and D.C.Y. Lim, 2001. A machine learning approach to coreference resolution of noun phrases. Comput. Ling., 27: 521-544.

Stoyanov, V., C. Cardie, N. Gilbert, E. Riloff and D. Buttler *et al*., 2010. Coreference resolution with reconcile. Proceedings of the ACL 2010 Conference Short Papers, July 11-16, 2010, Association for Computational Linguistics, Stroudsburg, Pennsylvania, USA, pp: 156-161.

Sutton, C. and A. McCallum, 2006. An Introduction to Conditional Random Fields for Relational Learning. In: Introduction to Statistical Relational Learning. Getoor, L. (Ed.). Cambridge University Press, Massachusetts, USA., pp: 93-128.

Vilain, M., J. Burger, J. Aberdeen, D. Connolly and L. Hirschman, 1995. A model-theoretic coreference scoring scheme. Proceedings of the 6th Conference on Message Understanding, November 6-8, 1995, Association for Computational Linguistics, Stroudsburg, Pennsylvania, USA, ISBN: 1-55860-402-2, pp: 45-52.

Yang, B., C. Cardie and P. Frazier, 2015. A hierarchical distance-dependent bayesian model for event coreference resolution. Trans. Assoc. Comput. Ling., 3: 517-528.