Co-reference Resolution in Tamil Text

Vijay Sundar Ram R. and Sobha Lalitha Devi
AU-KBC Research Centre,
MIT Campus of Anna University,
Chennai, India
{sundar,sobha}@au-kbc.org

Abstract

Natural Languages are cohesive. Cohesiveness is brought by various language phenomenons. Co-referring entities bind the sentence through reference phenomenon. These co-referring entities include various anaphoric expressions namely pronominals, reflexives, reciprocal, distributives, noun-noun anaphora and definite descriptions. These co-referring entities form the co-reference chains. In this work, we present a methodology to identify the co-reference chains in Tamil text. Evaluation of the system shows encouraging results.

1 Introduction

Cohesiveness of the text is brought by various language phenomenons. Co-referring entities play a crucial part in binding discourse intra and inter sententially. These co-referring entities form a chain in the text. The present work is on identifying the co-reference chains in Tamil text by identifying the co-referring entities. The co-referring entities consist of pronominal, reciprocal, reflexives, distributives, definite description and noun-noun anaphora and their antecedents. Co-reference chains are very essential in building cutting edge natural language processing tools such as profile building, entity based summary generator, entity specific sentiment analyser etc. Consider the following the discourse.

Ex. 1.a
raaju$_{1,2}$ ***avanutaiya$_{1}$*** *naNpan* ***baaluvin$_{2}$***
Ramu(N) he(PN)+gen friend(N) Balu(N)-gen
viittiRku *cenRaan.*
house(N)+dat go(V)+past+3sm
(Raju went to his friend Baalu's house.)

Ex. 1.b
ivarkaL$_{2,3}$ ***oruvarukkoruvar$_{3}$*** *nanku*
They(PN) eachother(N) very_well(ADJ)
aRivaarkaL. (1.1.b)
know(V)+past+3p
(They know each-other very well.)

Ex. 1.c
baaluvin *thaay* ***siiththaa$_{4}$***
Balu(N)-gen mother(N) Sita(N)
oru *aaciriyar.* (1.1.c)
one(ADJ) teacher(N)
(Baalu's mother Sita is a teacher.)

Ex. 1.d
baalu$_{4}$ ***raajuvai1*** ***than$_{4}$*** ***naNparkaL$_{5}$***
Balu(N) Raju(N)+acc his(PN) friends(N)
ovvoruvarukkum$_{5}$ aRimukappatuththinaan.
everone(PN) introduced(V)
(Baalu introduced Raju to every-one of his friends.)

There are various anaphoric expressions in the above discourse (Ex.1) and following are the pronominals, 1.'avanutaiya' [his] refers to 'raaju' (Ex.1.a), 2. 'than' [his] (Ex.1.d) refers to 'baalu' (Ex.1.d), 'ivarkaL' [they] in (Ex.1.b) refers to 'raaju' and 'baalu' present in (Ex.1.a) as two independent mentions. Here the antecedent of the pronoun 'ivarkaL' is two independent nouns 'raaju' and 'baalu'. This type of antecedents is known as split-antecedents.

The reciprocal 'oruvarukkoruvar' [each-other] in Ex.1.b refers to 'ivarkaL' [they], the subject of the sentence. In Ex.1.f, 'ovvoruvar' [every-one], Distributive refers to 'naNparkaL' (friends). The other type of anaphor is the noun-noun anaphor which is present in most of the sentences.

This can be seen in the above exampla as 'baalu' in Ex.1.d refers to 'baalu' in the previous sentence, 'baaluvin' in Ex.1.a. Similarly 'raajuvai' in Ex.1.d refers to 'raaju' in Ex.1.1.a. From

S Bandyopadhyay, D S Sharma and R Sangal. Proc. of the 14th Intl. Conference on Natural Language Processing, pages 392–401,
Kolkata, India. December 2017. ©2016 NLP Association of India (NLPAI)

the above explanation we can form the co-reference chain for each type of reference and they are given below.

Following are the Co-reference Chains from the example sentences from Ex.1.a to Ex.1.d.

- '**raaju**' (Ex.1.a) , '**avanutaiya**' [his] (Ex.1.a), '**raajuvai**' (Ex.1.d)
- '**naNparkaL**' (Ex.1.d), '**ovvoruvar'** [every-one] (Ex.1.d)
- '**raaju**', '**baalu**' (Ex.1.a), '**ivarkaL**' [they] (Ex.1.b), '**oruvarukkoruvar'** [each-other] (Ex.1.b)

Thus the anaphoric expressions such as Pronominal, Split-antecedents, Reciprocal, Reflexives, Distributive, One anaphora, Definite-Descriptions and Noun-Noun anaphora constitute the co-reference chains. In the present work, the co-reference chains are built by resolving these anaphoric entities.

Co-reference resolution was the shared task in DARPA's Message Understanding Coreference MUC-6 (1995) and MUC-7 (1997). These two shared tasks were the early initiatives which kick started machine learning based approach for co-reference relation resolution task. Aone & Bennet (1995), McCharthy & Lehnert (1995), Fisher et al. (1995) had used decision tree learning algorithm to come up with co-reference resolution system. . Aone & Bennet (1995) demonstrated the system with Japanese texts along with English texts. Kelher et al. (1997) used maximum entropy modelling technique to built co-reference resolution engine. Cardie & Wagstaff (1999) came up with an un-supervised learning approach to identify co-reference relation. They have evaluated their engine on MUC-6 dataset. Soon et al. (2000) used decision tree learning approach to identify the co-referencing pairs and used pair-wise model to build the co-reference chains. Ng & Cardie (2002) enhanced Soon et al. (2000) decision tree learning approach with more linguistic and heuristic features. They used best-first clustering methodology to build the co-reference chains. First-order probabilistic model was by Culcotta et al. (2007). Bengston and Roth (2007) tried to present that the approach by Soon et al. (2000) would perform better with better features. They re-implement it with modified features. Rahman & Ng (2011) employed cluster-ranking approach to perform co-reference resolution. A multilevel sieve based approach was performed by Raghunathan et al. (2010). SemEval (2010) Coreference Resolution in Multiple Languages aimed to explore the portability of systems across languages, need for different levels of linguistic information (Recasens, 2010).

The flow of the paper is as follows. In the following section, we present a brief introduction on characteristics of Tamil. We have explained our approach in the third section. In fourth section, we have presented on the experiment, result and observation. The paper concludes with the conclusion section.

2 Characteristics of Tamil

Tamil belongs to the South Dravidian family of languages. It is a verb final language and allows scrambling. It has post-positions, the genitive precedes the head noun in the genitive phrase and the complementizer follows the embedded clause. Adjective, participial adjectives and free relatives precede the head noun. It is a nominative-accusative language like the other Dravidian languages. The subject of a Tamil sentence is mostly nominative, although there are constructions with certain verbs that require dative subjects. Tamil has Person, Number and Gender (PNG) agreement. It is a relatively free word order language, but when it comes to noun phrases and clausal constructions it behaves as a fixed word order language. Clausal constructions are introduced by non-finite verbs. Tamil has copula drop, accusative drop, genitive drop, and PRO drop (subject drop).

3 Our approach

In this section, we present our approach to identify the co-reference chain in Tamil text. In most of the published works, single machine learning technique with a set of features is used. We have varied from other approaches by using different methodologies and features for resolution of various anaphoric expressions as resolution of pronominals, reciprocal, reflexives, distributives requires syntactic features and resolution noun-noun anaphora and definite description requires semantic features.

Our approach starts with preprocessing input text with sentence splitter, tokeniser and syntactic modules namely morphological analyser built using paradigm based approach (Sobha et. al, 2013), PoS tagger (Sobha et. al, 2016) and chunker using Conditional Random Fields (CRFs) technique, and clause boundary identifier built using CRFs with linguistic rules as features (Ram et. al., 2012) and Named Entity recognizer built using CRFs where statistical features are used (Malarkodi et. al., 2012).

The preprocessed text is fed to various anaphora resolution engines, pronominal resolution engine, where the singular pronouns are resolved using ML techniques and plural pronouns are resolved using salient weights based approach; reflexives, reciprocals and distributives are resolved using rule based approach; followed by noun-noun anaphora resolution and definite description identification using CRFS techniques. Using the different anaphors and their antecedents, co-reference chains are built using pair-wise clustering techniques with restriction rules. We have described the methodologies of resolution of various anaphoric expressions in the following sub-section.

3.1 Pronominal Resolution

We have performed the resolution of singular and plural pronouns with different techniques as plural pronouns can have plural noun phrase, co-ordinated noun phrases and split-antecedents.

Singular Pronoun Resolution: Singular pronoun resolution is built using Conditional Random Fields (CRFs) technique (Kudo,2013). Though CRFs is notable for sequence labelling task, we used this technique to classify the correct anaphor-antecedent pair from the possible candidate NP pairs by presenting the features of the NP pair and by avoiding the transition probability. While training we form positive pairs by pairing anaphoric pronoun and correct antecedent NP and negative pairs by pairing anaphoric pronouns and other NPs which match in person, number and gender (PNG) information and match Named entities (NE) constraints with the anaphoric pronoun. NE constraints check for the type of NE which can be the antecedent for a particular pronoun, such person pronoun can have Individual as antecedent and Location NE can never be its antecedent. These positive and negative pairs are fed to the CRFs engine and the language model is generated. While testing, when an anaphoric pronoun occurs in the sentence, the noun phrases which match in PNG and satisfies NE constraints with the pronoun, that occur in the preceding portion of the sentence and the four preceding sentences are collected and paired with the anaphoric pronoun and presented to CRFs engine to identify the correct anaphor-antecedent pair.

The features used in machine leaning technique are as follows.

Positional Features: Is the candidate antecedent occur in the same sentence where the pronoun has occurred or in the prior sentences.

Syntactic Argument: The case marker affixed to the NP helps in identifying the systactic argument of the sentence such as subject, object, indirect object, are obtained from the case suffix affixed with the noun phrase. The case marker information is available from the morphological analysers output.

Linguistic Characteristics:

a) PoS tag and chunk information of Candidate NP, suffixes affixed with the noun.
b) The suffixes which show the gender which gets attached to the verb.
c) Whether the candidate NP (probable antecedent) is Possessive.

Constraint Features: The constraint features are obtained from clause boundary information. If the pronoun is possessive, the nominative NP within the clause has a high probability to be the antecedent. If the pronoun is a non-possessive pronoun, the nominative NP in the immediate preceding clause has a high probability to be the antecedent. So we check the position of the candidate NP with respect to clause boundary such as whether the candidate NP occurs in current clause or immediate clause or non-immediate clause.

Plural Pronoun Resolution: Plural pronoun resolution engine is developed using a salience factor weights based approach. The antecedent for a plural pronoun can be a plural Noun phrase, co-ordinated NPs and Split antecedent.

We weigh each of the Noun phrase matching in gender with the plural pronoun. The features for the salience factors are obtained from the syntactic parsing output. We have mentioned the salience factors and its weights were as per Sobha (2007). Following is the algorithm used in resolving plural pronouns.

Step 1: If a plural pronoun occurs then Step 2.
Step 2: Collect all Noun phrases in the current sentence and previous four sentences which match with the gender of the plural pronoun.
Step 3: Each Noun phrase (NP) in the collection of possible antecedent set is scored with salience factor weights.
Step 4: The NPs re-sorted in descending order with their weights.

Step 5: If the highest scored NP is a plural NP, then it is selected as the Antecedent. Else step 6.
Step 6: If the highest scored NP is singular, check if this NP is part of co-ordinated NP or split antecedent, then choose the co-ordinated NP or the split antecedent as the antecedent.

Check for Co-ordinated NP**:** Co-ordinated NPs are those NPs which have the same scores as the highest score NP.

Check for Split-antecedents: We attempt to identify split-antecedents using selectional restriction rules of the verb, categorizing the nouns based on its sub-categorization information and ranking the possible antecedents using salience weights.

Sub-categorization features explain the nature of a noun. Sub-categorization information includes the features such as [±animate], [±concrete], [±edible] etc. The verbs describe the action or the process in the nature and this allow the verbs to take nouns with specific sub-categorization feature as its syntactic arguments. This is defined as the selectional restriction rules of a verb.

Ex.2
raam aappil caappittaan.
Ram(N) apple(N) eat(V)+past+3sn
(Ram ate an apple).

Here in Ex.2 'raam' (Ram) has the sub-categorization feature [+animate, +human] and 'aappil' (apple) with [+edible]. The selectional restriction features required by the verb 'caappitu' (eat) for selecting its subject and object are [+animate] and [+edible] respectively. If there is a violation in SR rules, the sentence can be syntactically correct but it will not be semantically correct. Verb has the right to select its arguments. We have grouped the verbs according to the sub-categorization information of the subject and object nouns. A group of commonly used 1500 verb senses are analyzed and 500 selectional restriction rules are derived by (Ananth and Sobha, 2010). The sub-categorization features of a noun are explained in the next section. A sample rule is shown in fig 1.

Using the selectional restriction rules and the sub-categorization information of nouns we try to group the noun phrases to form groups which can be possible split-antecedents.

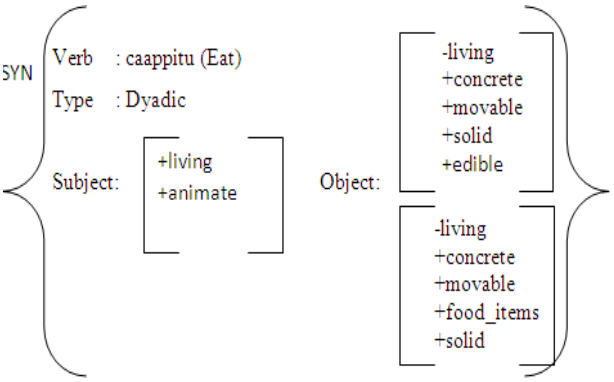

Figure 1. Selectional restriction rule for 'caapitu' (eat).

Following are the steps involved in identifying the split-antecedents. In the first step, we enrich the nouns and the verbs with its sub-categorization information, and selectional restriction rules respectively. The named entities (NEs) are mapped to the sub-categorization features, so we get the sub-categorization information using the NE information as described in the example Ex.3 and Ex.4.

Ex.3.
Person: [+living; +animate; +vertebrate; +mammal; +human;]
Ex.4.
Location: [-living; -moveable; +landscape]

In the second step, when a plural pronoun is encountered in the sentence, the preceding portion of the sentence and two preceding sentences are considered for analysis, as Gatt et al. (2009) has shown that the distance between plural pronouns and its antecedent are very few sentences away. The noun phrases in the preceding sentences are analysed and grouped to form the possible antecedents. For grouping the NPs, the NPs need to satisfy the following matching conditions.

a) The NPs can be grouped together if they have same sub-categorization information or till the last but one node in the ontology is same. Example [+living; +animate; +vertebrate; +mammal; +human; +female] and [+living; +animate; +vertebrate; +mammal; +human; -female] are considered to be same since both are same till last but one node and the exceptions are as follows:

In the case of NPs with sub-categorization [+living] and do not have [+human], we look for sub-categorization match between the NPs only till [+living; +animate] and such NPs are grouped together.

Following are the steps involved to form possible candidates by grouping the NPs.

1. Identify the plural pronoun in n^th^ sentence.
2. If the finite verb of the sentence having plural noun or plural possessive pronoun is followed by noun form of the verbs such as ‘inai’ (merge), ‘manam’ (marry), ‘vivaakaraththu_cey’ (disvorce), ‘kaathal’ (love) then look for two nouns which satisfy the sub-categorisation matching condition in preceding two sentences, group these two NPs as a possible split antecedent candidate.
3. Consider sentence n-2th, n-1th and in nth sentence consider the portion preceding to the plural pronoun to form a candidate sentence set.
4. For each sentence in the candidate sentence set
    a. Noun Phrases with conjunct suffix ‘um’ or conjunct word ‘maRRum’ (and) are united to form conjunct NPs.
5. For each sentence in sentence set
    a. If there exists NPs satisfying the matching condition, then the NPs are grouped together.
6. Group the NPs that occur in same syntactic argument position and satisfy the matching condition across nth, n-1th and n-2th sentences.

Table 1 Salience Factors and its Weights

S. No.	Salience Factors	Weights
1	Same Ontology Nodes	30
2	NPs with following verbs	30
3	NPs with same syntactic argument position	20
4	NPs with different syntactic argument position	10
5	NPs are syntactic argument for verbs having same SR rules	30
6	NPs are syntactic argument for verbs with different SR rules	10
7	NPs in current nth sentence	30
8	NPs in n-1th sentence	20
9	NPs in n-2th sentence	10

In the third step, when the possible antecedents are formed by grouping the NPs, they are ranked based on the salience factors derived from the features of NPs such as the sub-categorization information of NPs, the SR rules of verbs followed by the NPs and the syntactic argument position of the NPs in the sentences. The salience factor weights are described in table 1. The weights for the salience factors were initially manually assigned based on linguistic considerations and fine-tuned through experiments (Ram & Sobha, 2016).

Resolution of Reflexives: The antecedent of the reflexive is always the subject of the clause, where the reflexive occur. So the antecedent of the reflexive is identified with the rule based approach.

Resolution of Reciprocals and Distributives: Reciprocals and Distributives are handled similar to the reflexives. The antecedent of the Reciprocals and Distributives will the plural nominative noun phrase in the same clause. The resolution of the reciprocals and distributives are done using a rule based approach.

Noun-Noun Anaphora Resolution: Noun-Noun Anaphora resolution is the task of identifying the referent of the noun which has occurred earlier in the document. In a text, a noun phrase may be repeated as a full noun phrase, partial noun phrase, acronym, or semantically close concepts such as synonyms or superordinates. The engine to resolve the noun anaphora is built using Conditional Random Fields technique. Features used in Noun-Noun Anaphora Resolution are discussed below.

We consider the noun anaphor as NP_i and the possible antecedent as NP_j. Unlike pronominal resolution, Noun-Noun anaphora resolution requires features such as similarity between NP_i and NP_j.We consider word, head of the noun phrase, named entity tag and definite description tag, gender, sentence position of the NPs and the distance between the sentences with NP_i and NP_j as features.

Features used for ML

The features used in the CRFs techniques are presented below. The features are divided into two types.

Individual Features:

1. Single Word: Is NP_i a single word; Is NP_j a single word
2. Multiple Words: Number of Words in NP_i; Number of Words in NPj
3. PoS Tags: PoS tags of both NP_i and NP_j.
4. Case Marker: Case marker of both NP_i and NP_j.

5. Presence of Demonstrative Pronoun: Check for presence of Demonstrative pronoun in NP_i and NP_j.

Comparison Features

1. Full String Match: Check the root words of both the noun phrase NP_i and NP_j are same.
2. Partial String Match: In multi world NPs, calculate the percentage of commonality between the root words of NP_i and NP_j.
3. First Word Match: Check for the root word of the first word of both the NP_i and NP_j are same.
4. Last Word Match: Check for the root word of last word of both the NP_i and NP_j are same.
5. Last Word Match with first Word is a demonstrator: If the root word of the last word is same and if there is a demonstrative pronoun as the first word.
6. Acronym of Other: Check NP_i is an acronym of NP_j and vice-versa.

Definite Description Identification: Definite Description (DD) is a unique denoting phrase of an entity. Consider the example, Indian Prime Minister Narendra Modi. Here the phrase “Indian Prime Minister” describes about Person Entity ‘Narendra Modi’.

We used CRFs technique to identify the DD relations. We have used the PoS, NE information of the two NPs (possible definite description NP and Entity NP) and two preceding and following words as the feature to train the CRFs engine.

Co-reference Chain Builder: We used CRFs technique to identify the DD relations. We have used the PoS, NE information of the two NPs (possible definite description NP and Entity NP) and two preceding and following words as the feature to train the CRFs engine. We have used various constraint rules to generate the co-reference chains from the co-referring antecedent NP and anaphor NP pairs. We have built the constraint rules based on the types of the NPs in the co-referring NP pairs. Co-referring pairs obtained from different pronominal resolution engines are treated with high confidence. Co-referring NPs having exact match and not a partial NPs of any other NP, then these pair of NPs are considered for generating co-reference chains. If one of the NPs in the co-referring NP pair is a definite description, then the distance between should be checked. If it is close by in the same sentence then it is considered for co-reference chain generation. If one of the NP is a partial NP in the pair, then the distance between the partial NP and its co-referring NP is checked. If the distance is more than 3 sentences then the pair is dropped. If both the NPs are partial NPs and if the antecedent NP has a co-referring NP within proceeding three sentences then we can consider the pair for co-reference chain generation. The algorithm is presented as follows.

Algorithm for Generating Co-reference Chains

Step1: Type of NP in each co-referring NP pairs are identified.

Step2: For each of the identified co-referring NP pair; do step3 to

Step3: Check for the types of NPs in the co-referring NP pair,

If both the NPs have exact match and not a partial NP of the full NP then do step4.

If co-referring pair is obtained from the pronominal resolution engines, then do step4.

If one of the NP is a definite Description in the NP pair, check if the NPs occur close in the same sentence, then do step 4.

If the pair of NPs has a full NP and a partial NP, check if the NPs are in close proximity, i.e. within the three preceding sentences, then do step4.

If the pairs of NPs have both partial NPs, then check if the antecedent NP has a co-referring NP in the preceding three sentences, then do step4.

Step 4: Check if the NPs in the co-referring NPs are part of one of the existing clusters of co-referring NPs, then include these two pairs in that cluster. Else, introduce a new cluster with these two NPs.

Step 5: Each cluster is formed into a co-reference chain.

4 Experiment, Results and Discussion

We have manually annotated 1000 Tamil newwires collected from online Tamil web pages belonging to three domains, viz, sports, general and disaster. We had two annotators and the inter-agreement score is measured to be 0.78 kappa score. We have used 80% of the annotated corpus for developing the different anaphora resolution engines and co-reference chain builder. The rest 20% of the annotated corpus is used for testing the different anaphora resolution engines.

In the following table 2, we have presented the statistics of the annotated corpus.

Table 2: Basic Corpus Statistics

Details about Corpus	**Count**
Number of Web Articles annotated	1,000
Number of Sentences	22,382
Number of Tokens	272,415
Number of Words	227,615

Following table 3, has the statistics of the different anaphoric expression annotated in the corpus.

Table 3: Statistics of Anaphoric expressions in the Corpus

S.No	**Type**	**Number of Occurrence**
1	Noun-Noun Anaphora	11,935
2	Anaphoric Pronominal	4,160
3	Definite-Description	1,890
4	Reflexives	29
5	Reciprocal	31
6	Plural pronouns with split-antecedent	190
7	Distributives	8
	Total	18,243

The co-reference chains are evaluated with standard evaluation metrics such as MUC, B-Cubed, CEAFe, CEAFm and BLANC. The performance scores for co-reference chain identification are presented in table 4.

Table 4: Performance scores for Co-reference chains

S.No.	**Metric**	**Precision (%)**	**Recall (%)**	**F-Measure (%)**
1	MUC	51.21	35.5	41.94
2	B-CUB	74.8	52.71	61.84
3	CEAFm	46.31	46.31	46.31
4	CEAFe	30.2	44.73	36.06
5	BLANC	64.35	56.74	57.80
6	Average	53.37	47.19	48.79

The performance scores of various anaphora resolution modules with system preprocessed corpus and gold standard corpus as input is presented in table 5.

Table 5 presents the comparison of performance scores between the results obtained by giving preprocessed corpus, Gold standard and system processed, as input to the anaphoric systems. This brings out the inherent errors of each anaphora resolution systems and the errors introduced by preprocessing modules. On analysing the gold standard corpus result, we find pronominal resolution and one-anaphora resolution need improvement at the anaphora analysis level. The tendency of pronominal resolution engine to choose the first nominative as antecedent is one of the reason and this needs further analysis.

Table 5 Comparison of Results with System Preprocessed Corpus and Gold standard Corpus as Input

S. No	**Task**	**System Preprocessed Corpus**			**Gold Standard Corpus**		
		Precision (%)	**Recall (%)**	**F-Measure (%)**	**Precision (%)**	**Recall (%)**	**F-Measure (%)**
1	Singular Pronoun Resolution	79.04	62.87	70.03	81.63	75.39	78.38
2	Plural Pronoun	81.41	64.7	72.09	82.15	76.21	79.06
3	Reflexives	96.54	93.34	94.91	96.54	93.34	94.91
4	Reciprocals	98.17	97.39	97.78	98.17	97.39	97.78
5	Distributives	97.38	95.56	96.46	97.38	95.56	96.46

5	Definite-Description	92.98	70	79.87	93.83	78.56	85.51
6	Noun-Noun Anaphora Resolution	86.14	66.67	75.16	87.19	78.32	82.52

Table 6 Percentage Distribution of Errors introduced by Various Preprocessing Modules

S. No	Task	Intrinsic Errors of the anaphoric modules (%)	Total Percentage (%) of Error introduced by Preprocessing modules	Percentage of error contributed by Each Preprocessing module				
				Anaphoric Non anaphoric Identification (%)	Morphological Analyser (%)	PoS Tagger (%)	Chunker (%)	Named Entity Recogniser (%)
1	Singular Pronoun Resolution	21.62	8.35	10.86	26.14	40.65	22.35	
2	Plural Pronoun	20.94	6.97	12.56	27.44	37.23	22.77	
3	Reflexives	5.09	0		23	41.66	35.34	
4	Reciprocals	2.22	0		41.45	32.15	26.40	
5	Distributives	3.54	0		38.56	28.14	33.30	
6	Definite-Description	14.49	5.64			25.24	30.27	44.49
7	Noun-Noun Anaphora Resolution	17.48	7.36		11.56	18.78	36.44	33.22

In table 6, we have presented the percentage of intrinsic errors, the total percentage of errors introduced by preprocessing modules to each anaphora resolution engine and the percentage of errors contributed by each preprocessing modules to the total preprocessing errors.

With the informations from table 6, we can understand the importance of features derived from each preprocessing module for developing various anaphora resolution engines.

The output from the gold standard corpus as input is analysed and the observations are discussed below.

In singular pronominal resolution engine, which is built using CRFs techniques, the first nominative NP is choosen as the antecedent if the sentences have more than one nominative NP. Consider the following discourse.

Ex. 5.a.
munnaal thalaivar coomuvin
Formar(N) leader(N) Soomu(N)+pos
aatharavaalaraana raamu
supporter(N) Ramu(N)
neeRRu peecinaar.
yesterday(Adv) talk(V)+past+3sh

Ex.5.b.
avar kuuRiyathu.
He(PN) say(V)+past+3sn

The antecedent for ‘avar’ 3rd person singular honorofic pronoun in Ex.5.b is ‘raamu’ (Ramu) in Ex.5.a. But the resolution engine identifies ‘munnaal thalaivar’ (former leader) as the antecedent. This is also observed in plural pronoun resolution engine. Consider the following discourse.

Ex.6.a
puunaikaL miinkaL neeRRu caappittana.
Cat(N)+Pl fish(N)+Pl yesterday eat(V)+past+3pc
(Cats ate the fishes yestreday.)

Ex.6.b
avai nalla katal miinkaL.
They(PN) good(Adj) sea(N) fish(N)+Pl
(They are good sea fishes.)

Consider the discourse Ex.6. The plural neuter pronoun ‘avai’ in Ex.6.b, refers to ‘miiNkaL’ (fishes) in Ex.6.a. But the plural pronoun resolution engine identifies ‘puunaikaL’ (cats) which occur as the first NP in the sentence. Plural pronouns such as ‘naangkaL’ (we), ‘engkaL’ (our) occur in discourse with explicit antecedent in the discourse. The antecedent has to be understood as the group related to the speaker. These kinds are plural pronouns are not handled.

Noun-Noun anaphora resolution engine fails to handle definite NPs, as in Tamil we do not have definiteness marker, these NPs occur as common noun. Consider the following discourse.

Ex.7.a.
maaNavarkaL pooRattam katarkaraiyil
Student(N)+Pl demonstration(N) beach(N)+Loc
nataththinar.
do(V)+past+3pc
(The students did demonstartions in the beach.)

Ex.7.b.
kavalarkaL maaNavarkaLai kalainthu_cella
Police(N)+Pl students(N) disperse(V)+INF
ceythanar.
do(V)+past+3pc
(The police made the students to disperse.)

Consider the discourse Ex.7. Here in both the sentences ‘maaNavarkaL’ (students) has occurred referring to the same entity. But these plural NPs occur as a common nons and the definiteness is not signalled with any markers. So we have not handled these kinds of definite NPs which occur as common nouns.

5 Conclusion

We have presented a methodology to build co-reference chains in Tamil text. Co-reference chains are formed by the co-referential entities, which bring cohesiveness to the text. Co-referential entities include pronominals, pronouns with split-antecedents, reflexives, reciprocals, distributives, noun-noun anaphora and definite descriptions and their antecedents. Each of the anaphoric expressions is resolved using different methodologies as pronominal resolution requires syntactic features and noun-noun anaphora resolution and definite description identification requires semantic features. The co-reference chains are evaluated with standard metrics namely MUC, B-Cubed, CEAFe, CEAFm, and BLANC. The average precision is 53.37%, recall of 47.19% and F-measure of 48.79%.

Reference

Ananth Ramakrishnan, A & Sobha Lalitha Devi 2010, ‘An alternate approach towards meaningful lyric generation in Tamil’, Proceedings of the Workshop on Computational Approaches to Linguistic Creativity (CALC 2010), Association for Computational Lingusitics (ACL), LA, USA, pp. 31-39.

Aone, C & Bennett, S 1995, ‘Evaluating automated and manual acquisition of anaphora resolution strategies’. In: 33rd Annual Meeting of the Association for Computational Linguistics, pp. 122-129.

Bengtson, E & Roth, D 2008, ‘Understanding the value of features for coreference resolution’, In Proceedings of EMNLP, pp. 294-303.

Cardie, Claire & Kiri Wagstaff 1999, ‘Noun phrase coreference as clustering’, In Proceedings of the 1999 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora, pp. 82-89.

Culotta, A, Wick, M, Hall, R & McCallum, A 2007, ‘First-order probabilistic models for coreference resolution’, In Proceedings of HLT/NAACL, pp. 81-88.

Kehler Andrew 1997, ‘Probabilistic coreference in information extraction’, In Proceedings of the Second Conference on Empirical Methods in Natural Language Processing, pp. 163-173.

Kudo Taku. 2005. *CRF++, An Open Source Toolkit for CRF* [online] http://crfpp.sourceforge.net (accessed 3 January 2013).

Malarkodi C. S., Pattabhi R. K. Rao and Sobha Lalitha Devi. 2012, Tamil NER – Coping with Real Time Challenges, In Proceedings of Workshop on Machine Translation and Parsing in Indian Languages, COLING 2012, Mumbai, India

McCarthy, JF & Lehnert, WG 1995, ‘Using decision trees for coreference resolution’, In C. Mellish (Ed.), Fourteenth International Conference on Artificial Intelligence, pp. 1050-1055.

MUC-6 1995, Coreference task definition (v2.3, 8 Sep 95). In Proceedings of the Sixth Message Understanding Conference (MUC-6), pp. 335-344.

MUC-7 1997, Coreference task definition (v3.0, 13 Jul 97). In Proceedings of the Seventh Message Understanding Conference (MUC-7).

Ng, V & Cardie, C 2002, ‘Improving machine learning approaches to coreference resolution’, In. 40th Annual Meeting of the Association for Computational Linguistics, pp. 104-111.

Rahman, A & Ng, V 2011, ‘Narrowing the Modeling Gap: A Cluster-Ranking Approach to Coreference Resolution’, Journal of Artificial Intelligence Research, vol. 40, pp. 469-521 R.

Raghunathan, K, Lee, H, Rangarajan, S, Chambers, N, Surdeanu, M, Jurafsky, D & Manning, C 2010, ‘A multi-pass sieve for coreference resolution’, In Proceedings of EMNLP, pp. 492-501.

Ram, RVS, Bakiyavathi, T, Sindhujagopalan, R, Amudha, K & Sobha, L., 2012, ‘Tamil Clause Boundary Identification: Annotation and Evaluation’, In the Proceedings of 1st Workshop on Indian Language Data: Resources and Evaluation, Istanbul

Ram, RVS & Sobha Lalitha Devi 2016, ‘How to Handle Split Antecedents in Tamil?’, In proceedings of Coreference Resolution Beyond OntoNotes co-located with NAACL 2016, San Diego, California.

Recasens, M, M`arquez, L, Sapena, E, Mart´I, MA, Taul´e, M, Hoste, V, Poesio, M & Versley, Y 2010, ‘SemEval-2010 Task 1: Coreference Resolution in Multiple Languages’, In Proceedings of the 5th International Workshop on Semantic Evaluation, ACL 2010, Uppsala, Sweden, pp. 1-8.

Sobha, L 2007, ‘Resolution of Pronominals in Tamil. Computing Theory and Application’, The IEEE Computer Society Press, Los Alamitos, CA, pp. 475-79.

Sobha Lalitha Devi, Marimuthu K, Vijay Sundar Ram R, Bakiyavathi T and Amudha K. 2013, Morpheme Extraction in Tamil using Finite State Machines, In:Proceedings of Morpheme Extraction Task at FIRE 2013

Sobha Lalitha Devi, Pattabhi RK Rao and R Vijay Sundar Ram. 2016b, "AUKBC Tamil Part-of-Speech Tagger (AUKBC-TamilPoSTagger2016v1)". Web Download. Computational Linguistics Research Group, AU-KBC Research Centre, Chennai, India, 2016.

Soon WH Ng & Lim, D 2001, ‘A machine learning approach to coreference resolution of noun phrases’, Computational Linguistics, vol. 27, no. 4, pp. 521-544.