Proceedings of the WILDRE5– 5th Workshop on Indian Language Data: Resources and Evaluation, pages 20–24
Language Resources and Evaluation Conference (LREC 2020), Marseille, 11–16 May 2020

Handling Noun-Noun Coreference in Tamil

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

Abstract

Natural language understanding by automatic tools is the vital requirement for document processing tools. To achieve it, automatic system has to understand the coherence in the text. Co-reference chains bring coherence to the text. The commonly occurring reference markers which bring cohesiveness are Pronominal, Reflexives, Reciprocals, Distributives, One-anaphors, Noun–noun reference. Here in this paper, we deal with noun-noun reference in Tamil. We present the methodology to resolve these noun-noun anaphors and also present the challenges in handling the noun-noun anaphoric relations in Tamil.

Keywords: Tamil, noun-noun anaphors, Error analysis

1. Introduction

The major challenge in automatic processing of text is making the computer understand the cohesiveness of the text. Cohesion in text is brought by various phenomena in languages namely, Reference, Substitution, Ellipsis, Conjunction and Lexical cohesion (Halliday & Hasan 1976). The commonly occurring reference markers which bring cohesiveness are Pronominal, Reflexives, Reciprocals, Distributives, One-anaphors, Noun–noun reference. The coreference chains are formed using them. Coreference chains are formed by grouping various anaphoric expressions referring to the same entity. These coreference chains are vital in understanding the text. It is required in building sophisticated Natural Language Understanding (NLU) applications. In the present work, we focus on resolution of noun-noun anaphors, which is one of the most frequently occurring reference entities. A noun phrase can be referred by a shorten noun phrases or an acronym, alias or by a synonym words. We describe our machine learning technique based approach on noun-noun anaphora resolution in Tamil text and discussed the challenges in the handling the different types of noun-noun anaphora relations. We have explained noun-noun anaphora relation with the example below.

Ex 1. a
taktar apthul kalam *oru vinvezi*
Dr(N) Abdul(N) Kalam(N) one(QC) aerospace(N)

vinnaani.
scientist(N).
(Dr. Abdul Kalam was an aerospace scientist.)

Ex 1. b
kalam *em.i.ti-yil padiththavar.*
Kalam(N) M.I.T(N)+loc study(V)+past+3sh
(Kalam studied in MIT.)

Consider the discourse in Ex.1, 'taktar apthul kalam' (Dr. Abdul Kalam) in sentence Ex.1.a is mentioned as 'kalaam' (Kalam) in Ex.1.b.

One of the early works was by Soon et. al. (2001) where they have used Decision tree, a machine learning based approach for co-reference resolution. They have performed as pair-wise approach using Distance, String Match, Definite Noun phrase, Demonstrative noun phrase, both proper nouns, Appositives as features in the machine learning technique to resolve the noun-noun anaphors. Ng & Cardie (2002) extended Soon et. al. (2001) work by including lexical, grammatical, semantic, and PoS features. Culcotta et al. (2007) has performed first order probabilistic model for generating co-reference chain, where they have used WordNet, substring match as features to resolve the noun-noun relation. Bengston & Roth (2008) has presented an analysis using refined feature set for pair-wise classification. Rahman & Ng (2009) has proposed a cluster-ranking based approach. Raghunathan et. al (2010) has used multiple sieve based approach. Niton et al (2018) has used a deep neural network based approach. In the following section we have presented in the characteristics of Tamil, which make Noun-Noun anaphora resolution in Tamil a challenging task.

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.

Tamil 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. As in other languages, Tamil also has optional and obligatory parts in the noun phrase. Head noun is obligatory and all other constituents that precede the head noun are optional. Clausal constructions are introduced by non-finite verbs. Other characteristics of Tamil are copula drop, accusative drop, genitive drop, and PRO drop (subject drop). Clausal inversion is one of the characteristics of Tamil.

2.1 Copula Drop

Copula is the verb that links the subject and the object nouns usually in existential sentences. Consider the following example 2.

Ex 2: athu pazaiya maram. NULL
It(PN) old(ADJ) tree(N) (Coupla verb)
(It is an old tree).

The above example sentence (Ex.2.) does not have a finite verb. The copula verb ‘aakum’ (is+ past + 3rd person neuter), which is the finite verb for that sentence, is dropped in that sentence.

2.2 Accusative Case Drop

Tamil is a nominative-accusative language. Subject nouns occur with nominative case and the direct object nouns occur with accusative case marker. In certain sentence structures accusative case markers are dropped. Consider the following sentences in exaple.3

Ex3.
raman pazam caappittaan.
Raman(N) fruit(N)+(acc) eat(V)+past+3sm
(Raman ate fruits.)

In Ex.3, ‘raman’ is the subject, ‘pazaththai’ (fruit,N+Acc) is the direct object and ‘eat’ is the finite verb. In example Ex.3, the accusative marker is dropped in the object noun ‘pazam’.

2.3 Genitive Drop

Genitive drop can be defined as a phenomenon where the genitive case can be dropped from a sentence and the meaning of the sentence remains the same. This phenomenon is common in Tamil. Consider the following example 4.

Ex 4.
ithu raaman viitu.
(It)PN Raman(N) house(N).
(It is Raman’s house.)

In Ex.4, the genitive marker is dropped, in the noun phrase ‘raamanutiya viitu’ and ‘raaman viitu’ represents ‘raamanutiya viitu’ (Raaman’s house).

2.4 PRO Drop (Zero Pronouns)

In certain languages, the pronouns are dropped when they are grammatically and pragmatically inferable. This phenomenon of pronoun drop is also mentioned as ‘zero pronoun’, ‘null or zero anaphors’, ‘Null subject’.

These pose a greater challenge in proper identification of chunk boundaries.

3. Our Approach

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. These noun phrases mostly include named entity such as Individuals, place names, organisations, temporal expression, abbreviation such as ‘juun’ (Jun), ‘nav’(Nov) etc., acronyms such as ‘i.na’ (U.N), etc., demonstrative noun phrases such as ‘intha puththakam’ (this book), ‘antha kuuttam’ (that meeting) etc., and definite descriptions such as denoting phrases. The engine to resolve the noun anaphora is built using Conditional Random Fields (Taku Kudo, 2005) technique.

As a first step we pre-process the text with sentence splitter and tokenizer followed by processing with shallow parsing modules, namely, morphological analyser, Part of Speech tagger, Chunker, and Clause boundary identifier. Following this we enrich the text with Name Entities tagging using Named Entity Recognizer.

We have used a morphological analyser built using rule based and paradigm approach (Sobha et al. 2013). PoS tagger was built using a hybrid approach where the output from Conditional Random Fields technique was smoothened with rules. (Sobha et al. 2016). Clause boundary identifier was built using Conditional Random Fields technique with grammatical rules as features (Ram et al. 2012). Named Entity built using CRFs with post processing rules is used (Malarkodi and Sobha, 2012). Table1 show the precision and recall of these processing modules.

S.No.	Preprocessing Modules	Precision (%)	Recall (%)
1	Morphological Analyser	97.23	95.61
2	Part of Speech tagger	94.92	94.92
3	Chunker	91.89	91.89
4	Named Entity Recogniser	83.86	75.38
5	Clause Boundary Identifier	79.89	86.34

Table 1: Statistics of the Corpus.

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 in Noun-Noun Anaphora Resolution are discussed below.

3.1 Features used for ML

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

3.1.1 Individual Features

- Single Word: Is NPi a single word; Is NPj a single word
- Multiple Words: Number of Words in NPi; Number of Words in NPj
- PoS Tags: PoS tags of both NPi and NPj.
- Case Marker: Case marker of both NPi and NPj.
- Presence of Demonstrative Pronoun: Check for presence of Demonstrative pronoun in NPi and NPj.

3.1.2 Comparison Features

- Full String Match: Check the root words of both the noun phrase NP_i and NP_j are same.
- Partial String Match: In multi world NPs, calculate the percentage of commonality between the root words of NP_i and NP_j.
- First Word Match: Check for the root word of the first word of both the NP_i and NP_j are same.
- Last Word Match: Check for the root word of last word of both the NP_i and NP_j are same.

- 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.
- Acronym of Other: Check NP_i is an acronym of NP_j and vice-versa.

4. Experiment, Results and Evaluation

We have collected 1,000 News articles from Tamil News dailies online versions. The text were scrapped from from the web pages, and fed into sentence splitter, followed by a tokerniser. The sentence splitted and tokenised text is pre-processed with syntactic processing tools namely morphanalyser, POS tagger, chunker, pruner clause boundary identifier. After processing with shallow parsing modules we feed it to Named entity recogniser and the Named entities are identified. The News articles are from Sports, Disaster and General News.

We used a graphical tool, PAlinkA, a highly customisable tool for Discourse Annotation (Orasan, 2003) for annotating the noun-noun anaphors. We have used two tags MARKABLEs and COREF. The basic statistics of the corpus is given in table 2.

S.No	Details of Corpus	Count
1	Number of Web Articles annotated	1,000
2	Number of Sentences	22,382
3	Number of Tokens	272,415
4	Number of Words	227,615

Table 2: Statistics of the Corpus.

S. No.	Task	Precision (%)	Recall (%)	F-Measure (%)
1	Noun-Noun Anaphora Resolution	86.14	66.67	75.16

Table 3: Performance of Noun-Noun Anaphora Resolution

The performance scores obtained are presented in table 3. The engine works with good precision and poor recall. On analysing the output, we could understand two types of errors,1, errors introduced by the pre-processing modules and the intrinsic errors introduced by the Noun-noun anaphora engine. This is presented in table 4.

S. No	Task	Intrinsic Errors of the anaphoric modules (%)	Total Percentage (%) of Error introduced by Preprocessing modules
1	Noun-Noun Anaphora Resolution	17.48	7.36

Table 4: Details of errors

The poor recall is due to engine unable to pick certain anaphoric noun phrase such as definite noun phrases. In table 5, we have given the percentage of error introduced by different pre-processing tasks. We have considered the 7.38% error as a whole and given the percentage of contribution of each of the pre-processing tasks.

In noun-noun anaphora resolution, we consider Named entities, proper nouns, demonstrative nouns, abbreviations, acronyms, and try to identify their antecedents.

Percentage of error contributed by Each Preprocessing module			
Morphological Analyser (%)	PoS Tagger (%)	Chunker (%)	Named Entity Recogniser (%)
11.56	18.78	36.44	33.22

Table 5: Errors introduced by different pre-processing tasks

This task requires high accuracy of noun phrase chunker and PoS tagger. The errors in chunking and PoS tagging percolates badly, as correct NP boundaries are required for identifying the NP head and correct PoS tags are required for identifying the proper nouns. Errors in chunk boundaries introduce errors in chunk head which results in erroneous noun- noun pairs and correct noun-noun pairs may not be identified. The recall is affected due to the errors in identification of proper noun and NER.

Ex.5.a
aruN vijay kapilukku pathilaaka
Arun(N) vijay(N) Kapli(N)+dat instead

theervu_ceyyappattuLLar.
got_select(V)
(Instead of Kapil, Arun Vijay is selected)

Ex.5.b
vijay muthalil kalam iRangkuvaar.
He(PN) first(N)+loc groud(N) enter(V)+future+3sh
(He will be the opener.)

Ex.5.b has proper noun ‘vijay’ as the subject of the sentences and it refers to ‘aruN vijay’ (Arun Vijay), the subject of the sentence Ex.5.a. In Ex.5.a, chunker has tagged ‘aruN’, ‘vijay kapilukku’ as two NPs instead of ‘aruN vijay’ and ‘kapilukku’. Pronominal resolution engine has identifies ‘aruN’ as the referent of ‘avar’ instead of ‘aruN vijay’ in Ex.5.a. This is partially correct and full chunk is not identified due to the chunking error.

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.6.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.6.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.6. Here in both the sentences ‘maaNavarkaL’ (students) has occurred referring to the same entity. But these plural NPs occur as a common nouns and the definiteness is not signalled with any markers. So we have not handled these kinds of definite NPs which occur as common nouns.

Popular names and nicknames pose a challenge in noun-noun anaphora resolution. Consider the following examples; 'Gandhi' was popularly called as 'Mahatma', 'Baapuji' etc. Similarly 'Subhas Chandra bose' was popularly called as 'Netaji', 'Vallabhbhai Patel' was known as 'Iron man of India'. These types of popular names and nick names occur in the text without any prior mention. These popular names, nick names can be inferred by world knowledge or deeper analysis of the context of the current and preceding sentence. Similarly shortening of names such as place names namely 'thanjaavur' (Thanajavur) is called as 'thanjai' (Tanjai), 'nagarkovil' (Nagarkovil) is called as 'nellai' (Nellai), 'thamil naadu' (Tamil Nadu) is called as 'Thamilagam' (Tamilagam) etc introduce challenge in noun-noun anaphora identification. These shortened names are introduced in the text without prior mention. The other challenge is usage of anglicized words without prior mention in the text. Few examples for anglicized words are as follows, 'thiruccirappalli' (Thirucharapalli) is anglicized as 'Tirchy', 'thiruvananthapuram' (Thiuvananthapuram) is anglicized as 'trivandrum', 'uthakamandalam' is anglicized as 'ooty'. Spell variation is one of the challenges in noun-noun anaphora resolution. In News articles, the spell variations are very high, even within the same article. Person name such as 'raaja' (Raja) is also written as 'raaca'. Similarly the place name 'caththiram' (lodge) is also written as 'cathram'. In written Tamil, there is a practice of writing words without using letters with Sanskrit phonemes. This creates a major reason for bigger number of spell variation in Tamil. Consider the words such as 'jagan' (Jagan), 'shanmugam' (Shanmugam), and 'krishna' (Krishna), these words will also be written as 'cagan', 'canmugam' and 'kiruccanan'. These spell variations need to be normalised with spell normalisation module before pre-processing the text.

Spelling variation, Anglicization, Spelling error in NEs lead to errors in correct resolution of noun anaphors. Consider the following example, same entity 'raaja' (Raja) will be written as 'raaja' and 'raaca'.

Due to incorrect chunking, the entities required to form the co-refernce chains are partially identified. Consider example 7.

Ex.7
netharlaanthu aNi, netharlaanthu, netharlaanthu aNi
Netherland Team, Netherland, Netherland Team

Consider Ex.7, the same entities as occurred as both 'netharlaanthu aNi' (Netherland Team) and 'netharlaanthu' (Netherland) in the News article. The chunker has wrongly tagged 'netharlaanthu' (Netherland) and 'aNi' (team) as two different chunks. The resultant co-reference chain was 'netharlaanthu', 'netharlaanthu' and 'netharlaanthu'. 'aNi' in both NPs are missed out but to the chunker error.

Similarly in News articles, the place name entities are mentioned as place name or a description referring to the place name. Consider the following examples Ex.8.a, and Ex.8.b.

Ex.8.a
*mumbai, inthiyaavin varth** thalainakaram*
Mumbai, India's Economic Capital

Ex.8.b
kaaci, punitha nakaram
Kasi, the holy city

In Ex.8.a and Ex.8.b, there are two entities each in both and the NPs refer to the same entity. These kinds of entites are not handled by the Noun-Noun anaphora resolution engine and these entities are missed, while forming the co-reference chain. There are errors in identifying synonymous NP entities as presented in following discourse 9.

Ex.9.a
makkaL muuththa kaavalthuRaiyinarootu
People(N) senior(Adj) police(N)+soc

muRaiyittanar.
argue(V)+past+3p
(People argued with the senior police officer.)

Ex.9.b
antha athikaariyin pathiLai eeRRu
That(Det) officer(N)+gen answer(N) accept(V)+vbp

cenRanar.
go(V)+past+3p
(Accepting the officer's answer they left.)

Consider Ex.9.a and Ex.6.9.b, 'muuththa kaavalthuRaiyinarootu' (Senior police person) in Ex.9.a and 'athikaari' (officer) in Ex.9.b refer to the same entity. For robust Identification of these kinds of synonyms NPs we require synonym dictionaries.

Thus these kinds of noun phrases pose a challenge in resolving noun –noun anaphors.

5. Conclusion

We have discussed development of noun-noun anaphor resolution in Tamil using Conditional Random Fields, a machine learning technique. We have presented in detail, the characteristics of Tamil, which pose challenges in resolving these noun-noun anaphors. We have presented an in-depth error analysis describing the intrinsic errors in the resolution and the errors introduced by the pre-processing modules.

6. Bibliographical References

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

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

Halliday, M.A.K. and Hasan, R. (1976). Cohesion in English. Longman Publishers, London.

Malarkodi CS. Pattabhi RK Rao & 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.

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

Bartłomiej Niton, Paweł Morawiecki, and Maciej Ogrodniczuk. (2018). Deep Neural Networks for Coreference Resolution for Polish. In Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC) 2018, pp. 395-400.

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.

Rahman, A & Ng, V (2009). Supervised Models for Coreference Resolution. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp. 968-977.

Ram, RVS. Bakiyavathi, T. Sindhujagopalan, 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.

Orasan, C. (2003). PALinkA: A highly customisable tool for discourse annotation. In Proceedings of the 4th SIGdial Workshop on Discourse and Dialogue, ACL'03. pp. 39-43.

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

Sobha Lalitha Devi, Pattabhi RK Rao & Vijay Sundar Ram, R (2016). AUKBC Tamil Part-of-Speech Tagger (AUKBC-TamilPoSTagger 2016v1). Web Download. Computational Linguistics Research Group, AU-KBC Research Centre, Chennai, India

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.

Taku Kudo (2005). CRF++, an open source toolkit for CRF, http://crfpp.sourceforge.net