Handling Noun-Noun Coreference in Tamil

Vijay Sundar Ram and Sobha Lalitha Devi

AU-KBC Research Centre
Anna University, Chennai, India

Outline

- Objective
- Co-reference chain
- Our Approach
- Experiment and Results
- Error Analysis
- Intrinsic Errors

Objective

- Co-reference chains bring coherence
- Reference markers which bring cohesiveness
  - Pronominal, Reflexives, Reciprocals, Distributives, One-anaphors, Noun–noun reference
- Focus on resolution of noun-noun anaphors in Tamil
- Challenges in resolving it in Tamil

Co-reference Chains

- Coreference chains are formed by grouping various anaphoric expressions referring to the same entity.

- Early work in Co-reference resolution using ML
  - Soon et al (2000)
- Different ML Approaches
  - Decission Tree
  - First order probabilistic model
  - Multiple sieve based approach
  - Deep neural network based approach

Characteristics of Tamil

- South Dravidian family of language
- Relatively free word order language
- Verb final language and allows scrambling
- Nominative-accusative language
- Has Person, Number and Gender (PNG) agreement
- Clausal constructions are introduced by non-finite verbs.
- Copula drop, Accusative drop, Genitive drop, and PRO drop (Subject drop)

Our Approach

- Noun-Noun Anaphors
  - task of identifying the referent of the noun which has occurred earlier in the document.
  - Noun phrase may be repeated as a full noun phrase, partial noun phrase, acronym, or semantically close concepts such as synonyms or superordinates.
  - Named entities, Acronyms, Demonstrative noun phrases Definite descriptions

Our Approach(Contd...)

- Machine Learning Technique
  - Conditional Random Fields
- Data preparation:
  - Training data:
    - Positive and negative pairs of NPs (NP_i and NP_j)
  - Testing data:
    - pairs of NPs (NP_i and NP_j)
- Pre-processng of data:
  - Processed with morphological analyser, Part of Speech tagger, Chunker, Clause boundary identifier and Named Entity Recognizer.

Our Approach(Contd...)

- Features Uesd: 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.
- NE Category :
  - Named Entity tags of both NPi and NPj.
- Presence of Demonstrative Pronoun:
  - Check for presence of Demonstrative pronoun in NPi and NPj.

Our Approach(Contd...)

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.

Experiment and Evaluation

- Collected 1,000 News articles from Tamil News dailies online

- Preprocessed and Noun-Noun anaphoric relations are tagged using PALinkA tool
- Statistics of Corpus

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

Result and Analysis

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

Errors

- Intrinsic Errors of the Noun-Noun resolution Engine

S. No	Intrinsic Errors (%)
1	17.48

Errors due to Preprocessing modules

Considering the total errors as 100%

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

Ex.a

aruN vijay kapilukku pathilaaka *theervu_ceyyappattuLLar.*

Arun(N) vijay(N) Kapli(N)+dat instead select(V+past)

(Instead of Kapil, Arun Vijay is selected)

Ex.b

vijay muthalil kalam iRangkuvaar.

Vijay(N) first(N)+loc groud(N) enter(V)+future+3sh

(Vijay will be the opener.)

System output: vijay kapilukku ,vijay

Intrinsic Errors

- Fails to handle definite NPs,
  - no definiteness marker, these NPs occur as common noun.

Ex.a

maaNavarkaL *pooRattam* *katarkaraiyil* nataththinar.

Student(N)+Pl demonstration(N) beach(N)+Loc do(V)+past+3pc

(The students did demonstartions in the beach.)

Ex.b

kavalarkaL maaNavarkaLai kalainthu_cella ceythanar.

Police(N)+Pl students(N) disperse(V)+INF do(V)+past+3pc

(The police made the students to disperse.)

Here in both the sentences ‘maaNavarkaL’ (students) has occurred referring to the same entity.

Intrinsic Errors

Challenge in noun-noun anaphora resolution

- Popular names and nicknames
  - Gandhi’ -> ‘Mahatma’, ‘Baapuji’; ‘Subhas Chandra bose’ -> ‘Netaji’;
  - ‘Vallabhbhai Patel’ -> ‘Iron man of India’.
- Shortening of names
  - ‘thanjaavur’ (Thanajavur)-> ‘thanjai’ (Tanjai),
  - ‘nagarkovil’ (Nagarkovil) ->nellai’ (Nellai),
- Usage of anglicized words
  - ‘thiruccirappalli’ (Thirucharapalli) -> ‘Tirchy’, ‘thiruvananthapuram’ (Thiuvananthapuram) -> ‘trivandrum’, ‘uthakamandalam’ -> ‘ooty’.

Intrinsic Errors

Challenge in noun-noun anaphora resolution

- Spell variations
  - ‘raaja’ (Raja) -> ‘raaca’.
  - Person names are usually written in different spelling
- Named Entities mentioned with description referring to the Named Entities

mumbai, inthiyaavin varththaka thalainakaram

Mumbai, India’s Economic Capital

kaaci, punitha nakaram

Kasi, the holy city

Intrinsic Errors

Challenge in noun-noun anaphora resolution

Errors in identifying synonymous NP entities
Ex.a

makkaL muuththa kaavalthuRaiyinarootu *muRaiyittanar.*

People(N) senior(Adj) police(N)+soc argue(V)+past+3p

(People argued with the senior police people.)

Ex.b

antha athikaarikaLiyin pathiLai eeRRu cenRanar.

That(Det) officer(N)+PL+gen answer(N) accept(V)+vbp go(V)+past+3p

(Accepting the officer's answer they left.)

KaavalthuRaiyinarootu, athikaarikaL refers to the same entity.

Thank you