Identifying Participant Mentions and Resolving their Coreferences in Legal Court Judgements

Ajay Gupta⋆[2], Devendra Kumar Verma⋆[2], Sachin Pawar[1], Sangameshwar Patil[1], Swapnil Hingmire[1], Girish K. Palshikar[1], and Pushpak Bhattacharya[3]
{ajaygupta,devendrakuv,pb}@cse.iitb.ac.in
{sachin7.p,sangameshwar.patil,swapnil.hingmire,gk.palshikar}@tcs.com

[1]TCS Research, Tata Consultancy Services, Pune-411013, India

[2]Dept. of CSE, Indian Institute of Technology Bombay, Mumbai-400076, India

[3]Indian Institute of Technology Patna, Patna-801103, India

Abstract. Legal court judgements have multiple participants (e.g. judge, complainant, petitioner, lawyer, etc.). They may be referred to in multiple ways, e.g., the same person may be referred as lawyer, counsel, learned counsel, advocate, as well as his/her proper name. For any analysis of legal texts, it is important to resolve such multiple mentions which are coreferences of the same participant. In this paper, we propose a supervised approach to this challenging task. To avoid human annotation efforts for Legal domain data, we exploit ACE 2005 dataset by mapping its entities to participants in Legal domain. We use basic Transfer Learning paradigm by training classification models on general purpose text (news in ACE 2005 data) and applying them to Legal domain text. We evaluate our approach on a sample annotated test dataset in Legal domain and demonstrate that it outperforms state-of-the-art baselines.

Keywords: Legal Text Mining, Coreference Resolution, Supervised Machine Learning

1 Introduction

The legal domain is a rich source of large document repositories such as court judgements, contracts, agreements, legal certificates, declarations, affidavits, memoranda, statutory texts and so forth. As an example, the FIRE legal corpus[1] contains around 50,000 Supreme Court Judgements and around 80,000 High Courts judgements in India. Legal documents have some special characteristics, such as long and complex sentences, presence of various types of legal argumentation, and use of legal terminology. Legal document repositories are used for many

⋆ This work was carried out during the internship at TCS Research, Pune. Both the authors contributed equally.

[1] https://www.isical.ac.in/~fire/2014/legal.html

purposes, such as retrieving facts [18], [17], case summarization [23], precedence identification [8], identification of similar cases [9], extracting legal argumentation [12], case citation analysis [24] etc. Several commercial products, such as eBrevia, Kira, LegalSifter, and Luminance provide such services to lawyers.

A basic step in information extraction from legal documents is the extraction of various participants involved, say, in a court judgement. We define a *participant* as an entity of type person (PER), location (LOC), or organization (ORG). Typically, a participant initiates some specific action, or undergoes a change in some property or state due to action of another participant. Participants of type PER can be appellants, respondents, witness, police officials, lawyers, judge etc. Often organizations and locations play important roles in legal documents, and hence we include them as participants. For example, in `"...an industrial dispute was raised by the appellant, which was referred by the Central Government to the Industrial Tribunal ..."`, the two organizations mentioned (`"the Central Government", "the Industrial Tribunal"`) are participants.

Table 1. Sample text fragment from a court judgement. All the mentions of i^{th} participant are coreferences of each other and are marked with P_i.

[The respondent, Selvamuthukani]$_{P_1}$ is [the original complainant]$_{P_1}$. [The complainant]$_{P_1}$ alleged that [she]$_{P_1}$ was married to [Mr. Kannan, the accused]$_{P_2}$ on 16.6.1980 . According to [Mr. Singh, counsel for the complainant]$_{P_3}$, during the subsistence of [her]$_{P_1}$ marriage with [the accused]$_{P_2}$, [he]$_{P_2}$ married again with [K. Palaniammal]$_{P_4}$.

The same participant is often mentioned in many different ways in a document; e.g., a participant `Mr. Kannan` may be variously referred to as `the accused, he, her husband` etc. All such mentions of a single participant are coreferences of each other. Grouping all mentions of the same participant together is the task of *coreference resolution*. Many legal application systems provide an interactive, dialogue-based interface to users. Information extracted from legal documents, particularly about various participants and coreferences among them, is crucial to understand utterances in such dialogues; e.g., `What are the names of the accused and his wife?` in Table 1. In practice, we often find that a standard off-the-shelf coreference resolution tool fails to correctly identify all mentions of an participant, particularly on legal text [20]. Typically, a mention is not linked to the correct participant (e.g. Stanford CoreNLP 3.7.0 Coreference toolkit does not link `Selvamuthukani` and `The complainant`), or a mention is undetected (e.g. `the accused` is not detected as a mention) and hence, not linked to any participant. Nominal mentions consisting of generic NPs[2] (e.g., `complainant, prosecution witness`) are often not detected at all as participants or they are detected as participants but not linked to the correct participant mention(s).

[2] Noun Phrases with common noun as headword

We define *basic* mention of a participant to be a sequence of proper nouns (e.g., `K. Palaniammal`, `Mr. Kannan`), a pronoun (e.g., `he`, `her`) or a generic NP (e.g., `the complainant`). A basic mention can be either *dependent* or *independent*. A basic mention is said to be *dependent* if its governor in the dependency parse tree is itself a participant mention; otherwise it is called as *independent* mention. An independent mention can be basic (if it does not have any dependent mentions); otherwise a *composite mention* is created for it by recursively merging all its dependent mentions. For example, `Mr. Singh, counsel for the complainant` contains three basic participant mentions: `Mr. Singh`, `counsel` and `the complainant`. Here, only `Mr. Singh` is an independent mention and others are dependent mentions. The corresponding composite mention is created as `Mr. Singh, counsel for the complainant`.

In this paper, we focus on coreference resolution restricted to participants which consists of following steps: (i) identify basic participant mentions; (ii) merge dependent mentions into corresponding independent participant mentions to create composite mentions; and (iii) group together all independent participant mentions which are coreferences of each other. For step (i), we use a supervised approach, in which we train a classifier on a well-known labelled corpus (ACE 2005 [22]) of general documents to identify participants. We then use the learned model to identify participant mentions in legal documents. Then we have developed a rule-based system to perform step (ii). Finally, for step (iii) we use supervised classifier (such as Random Forest, SVM) . We evaluate our approach on a corpus of legal documents (court judgements) manually labelled with participant mentions and their coreference groups. We empirically demonstrate that our approach performs better than state-of-the-art baselines, including well-known coreference tools, on this corpus.

2 Related Work

The problem of coreference resolution specifically for Legal domain has received relatively limited attention in literature. The literature broadly categorized into two streams. One focuses on anaphora resolution [2] and the other addresses the problem of Named Entity Linking. *Anaphora Resolution* is a sub-task of *Coreference Resolution* where the focus is to find an appropriate antecedent noun phrase for each pronoun. The task of Named Entity linking [7, 4, 5] focuses on linking the names of persons / organizations and Legal concepts to corresponding entries in some external database (e.g. Wikipedia, Yago). In comparison, our approach focuses on grouping all the corefering mentions together including generic NPs.

Even in the general domain, the problem of coreference resolution remains an open and challenging problem [13]. Recently, Peng et al. [14, 15] have proposed the notion of Predicate Schemas and used Integer Linear Programming for coreference resolution. In terms of problem definition and scope, our work is closest to them as they also focus on all three types of mentions, i.e. named entities, pronouns and generic NPs.

3 Our Approach

We propose to use supervised machine learning approach to identify and link participant mentions in court judgements. Since there is lack of labeled training data in legal domain for this task, we map the entity mentions and coreference annotations in the

ACE 2005 dataset to suit our requirement. Table 2 gives overview of the proposed approach. Unlike a corpus annotated for traditional NER task, ACE 2005 dataset labels mentions of all 3 types which are of our interest in this paper, viz. proper nouns, pronouns and generic NPs. Hence, we found that ACE dataset can be adapted easily for this task with minor transformations.

The specific transformations required to ACE dataset are as follows – ACE provides annotations for 7 entity types: PER (person), ORG (organization), LOC (location), GPE (geo-political entity), FAC (facility), WEA (weapon) and VEH (vehicle). As our definition of participant only includes mentions of type PER, ORG and LOC, we ignore the mentions labelled with WEA and VEH. Also we treat LOC, GPE and FAC as a single LOC entity type. Moreover, we define basic participant mentions to be base NPs whereas ACE mentions need not be base NPs; e.g., for the base NP `the former White House spokesman`, ACE would annotate two different mentions: `White House` as ORG and `spokesman` as PER. However, we note that `spokesman` is the headword of this NP and other constituents of the NP (such as `the, White House`) are modifiers of this headword. So we expand this mention as a single basic participant mention of type PER. We converted the original ACE mention and coreference annotations accordingly.

Table 2. Overview of our approach

- **Phase-I: Training**
  Input: D : ACE 2005 corpus
  Output: $\mathcal{C}_M$: mention detector, $\mathcal{C}_P$: pair-wise coreference classifier
  T.1) Train $\mathcal{C}_M$ on D using CRF to detect participant mentions from a text.
  T.2) Train $\mathcal{C}_P$ on D using a supervised classification algorithm to predict whether participant mentions within a pair are coreferents.
- **Phase-II: Application**
  Input: d : test document, $\mathcal{C}_M$: mention detector, $\mathcal{C}_P$: pair-wise coreference classifier
  Output: $G = \{g_1, g_2, \ldots, g_k\}$: a set of coreference groups in d
  A.1) Let $\mathcal{M}$ be the set of entity mentions in d detected using $\mathcal{C}_M$.
  A.2) For each independent mention $m_i \in \mathcal{M}$
  merge its all dependent mentions recursively and remove them from $\mathcal{M}$.
  A.3) For each candidate pair of mentions $< m_i, m_j >$ in $\mathcal{M}$
  use $\mathcal{C}_P$ to classify whether m_i and m_j are coreferences of each other.
  A.4) Let G be the partition of $\mathcal{M}$ such that each $g_i \in G$ represents a group of coreferent mentions through transitive closure.

The three major steps in our approach are explained below in detail.

3.1 Identifying basic mentions of participants (T.1/A.1 in Table 2)

We model the problem of identifying basic mentions of participants as a sequence labeling problem. Here, similar to traditional Named Entity Recognition (NER) task, each word gets an appropriate label as per BIO encoding (**B**egin-**I**nside-**O**utside coding scheme used in NER). But unlike NER, we are also interested in identifying mentions in the form of pronouns and generic NPs. We employ Conditional Random Fields[3] (CRF) [10] for the sequence labeling task. Various features used for training the CRF model are described in Table 3.

Table 3. Features used by CRF for detecting basic mentions of participants

Feature Type	**Details**
Lexical	Word itself; lemma of the word; next and previous words
POS	Part-of-speech tags of the word as well as its previous and next words
Syntactic	Dependency parent of the word; Dependency relation with the parent
NER	Entity type assigned by the Stanford CoreNLP NER tagger
WordNet	WordNet hypernym type feature, derived from the hypernym tree, which can take one of $\{PER, ORG, LOC, NONE\}$ (e.g., for `complainant`, we get the synset (`person, individual, someone,` ...), which is one of the pre-defined synsets indicating PER, as an ancestor in the hypernym tree. For each participant type, we have identified such pre-defined synsets.)

3.2 Identifying independent participant mentions (A.2 in Table 2)

Our notion of independent mentions is syntactic, i.e. derived from the dependency parse tree. A basic participant mention is said to be *independent* if its dependency parent (with dependency relation type `nmod` or `appos`) is not a basic participant mention itself, otherwise it is said to be a *dependent* mention. In this stage, we merge all the *dependent* participant mentions (predicted in the previous step) recursively with their parents until only independent mentions remain.

3.3 Classifying mention pairs (T.2/A.3 in Table 2)

We model the coreference resolution problem as a binary classification task where each mention pair is considered as a positive instance iff the mentions are coreferences of each other. To generate candidate mention pairs, we consider threshold of 5 sentences. For this classifier, we derived 36 features using the dependency and constituency parse trees. The detailed description of the features is given in Table 4. Some of these features are based on the traditional mention-pair models in the literature [13, 6, 1, 19]. We have added some more features like: whether both the mentions are connected through a copula verb, whether both the mentions appear in conjunction, etc.

A binary classifier model is trained on the ACE dataset (T.2 in Table 2) and this model is used to classify candidate mention pairs (using the predicted participant mentions from A.2) in legal dataset (A.3 in Table 2). Here we have used transfer learning by training a model on ACE dataset and testing it on Legal dataset. We explored four different classifiers: Random Forest, SVM, Decision Trees, and Naive Bayes Classifier.

[3] We used CRF++ (`https://taku910.github.io/crfpp/`)

Table 4. Feature types used by the mention pair classification

Real-valued feature types:
(i) String similarity between the two mentions in terms of Levenshtein distance; (ii) No. of sentences / words / other mentions between the two mentions; (iii) Difference between the lengths of the mentions; (iv) Cosine similarity between word vectors (Google News word2vec embeddings) of head words of the mentions
Binary feature types:
(i) Whether both the mentions have same gender / number / participant type / POS tag; (ii) Whether both the mentions are in the same sentence; (iii) Whether any other mention is present in between; (iv) Whether both the mentions indefinite or definite; (v) Whether first / second mention is indefinite; (vi) Whether first / second mention is definite; (vii) Whether both the mentions are connected through a copula; (viii) Whether both the mentions appear in conjunction; (ix) Whether both the mentions are *nominal subjects* of some verbs; (x) Whether only the first or second mention is *nominal subject* of some verb; (xi) Whether both the mentions are *direct objects* of some verbs; (xii) Whether only the first or second mention is *direct object* of some verb; (xiii) Whether one mention is *nominal subject* and other is *direct object* of a same verb; (xiv) Whether both the mentions are pronouns; (xv) Whether only first or second mention is pronoun; (xvi) Whether first or second mention occurs at the start of a sentence

3.4 Clustering similar mentions (A.4 in Table 2)

To create the final coreference groups, we need to cluster the mentions using the output of classifier in step A.3. This is necessary because the pair-wise classification output in A.3 may violate the desired transitivity property [13] for a coreference group. We use clustering strategy similar to single-linkage clustering. We take the coreference mention pair classifier output as input to clustering system and process each Court Judgement output from coreference mention pair one by one. We select the mention pairs which are positive predicted examples from the input. These are called coreference pairs. These coreference pairs are used to create the coreference group as follows:

1. Select the coreference pairs one by one, check if they are present in the already created coreference groups.
2. If both are not present in any of the coreference groups, then create the new group by adding the both mentions from the mention pair.
3. If one is present in any of the already created coreference groups, add the second mention from the mention pair into that coreference group.
4. If both are present in any of the already created coreference group, do not add them into any coreference group.
5. Once all the mention pairs from a document are processed. We merge the disconnected coreference groups as follows :
   (a) Take the pair of coreference groups and check whether they are disjoint.
   (b) If they are disjoint, keep them as separate coreference groups.
   (c) If they are not disjoint, then merge those two coreference groups into single coreference group.

4 Experimental Analysis

We evaluate our approach on 14 court judgements: 12 judgements from The Supreme Court of India and 2 judgements from The Delhi High Court in the FIRE legal judgement corpus. On an average, a judgement contains around 45 sentences and 25 distinct participants. We manually annotated these judgements by identifying all the independent participant mentions and grouping them to create coreference groups.
Baselines: B1 is a standard baseline approach which uses Stanford CoreNLP toolkit. Here, basic participant mentions are identified as a union of the named entities (of type PER, ORG and LOC) extracted by the Stanford NER and the mentions extracted by the Stanford Coreference Resolution. Dependent mentions are merged with corresponding independent mentions by using the same rules as described in the step A.2 in Table 2. Final groups of coreferant participant mentions are then obtained by using coreference groups predicted by the Stanford Coreference toolkit. B2 is the state-of-the-art coreference resolution system based on Peng et al. [14, 15]. Unlike B1 and B2, our approach focuses on identifying coreferences only among the participant mentions and not ALL mentions. Hence, we discard non-participant mentions and coreference groups consisting solely of non-participant mentions from the predictions of B1 & B2.
Evaluation: We evaluate the performance of all the approaches at two levels: all independent participant mentions and clusters of corefering participant mentions. We use the standard F1 metric to measure performance of participant mention detection. For evaluating coreferences among the predicted participant mentions, we used the standard evaluation metrics [16], MUC [21], BCUB [3], Entity-based CEAF (CEAFe) [11] and their average. Table 5 shows the relative performance of our approach compared to the two baselines. Out of multiple classifiers Random Forest (RF) with Gini impurity as splitting criteria and 5000 trees provides the best result.

Table 5. Experimental results (RF: Random Forest, SVM: Support Vector Machines, DT: Decision Tree, NBC: Naive Bayes Classifier)

Algorithm	Participant mention			Canonical mentions									
				MUC			BCUB			CEAFe			Avg.
	P	R	F	P	R	F	P	R	F	P	R	F	F
B1	63.1	43.4	50.3	64.4	40.31	48.3	45.5	26.3	31.9	22.1	22.0	21.7	34.0
B2	64.5	41.1	46.5	62.0	31.4	38.0	52.8	20.1	25.3	24.8	28.9	25.0	29.4
RF	69.8	70.7	**70.2**	59.5	52.4	55.1	66.0	53.9	**58.1**	35.3	42.1	37.7	**50.3**
SVM				59.3	45.1	50.7	68.9	45.8	53.7	32.5	47.4	**37.9**	47.4
DT				54.8	69.5	**60.6**	31.4	74.1	42.4	30.6	16.9	21.0	41.3
NBC				54.6	50.5	52.1	57.0	51.8	53.0	34.1	38.4	35.5	46.9

5 Conclusion

This paper demonstrates that off-the-shelf coreference resolution does not perform well on domain-specific documents, in particular on legal documents. We demonstrate that using domain and application specific characteristics, it is possible to improve performance of coreference resolution. Identifying participant mentions and grouping their coreferents together is a challenging task in Legal text mining and Legal dialogue systems. We proposed a supervised approach for addressing this challenging task. We adapted ACE 2005 dataset by mapping its entities to participants in Legal domain. We showed that the approach outperforms the state-of-the-art baselines. In future, we plan to employ advanced transfer learning techniques to improve performance.

References

1. Samarth Agrawal, Aditya Joshi, Joe Cheri Ross, Pushpak Bhattacharyya, and Harshawardhan M Wabgaonkar. Are word embedding and dialogue act class-based features useful for coreference resolution in dialogue? In *Proc. of PACLING*, 2017.
2. K. Al-Kofahi, B. Grom, and P. Jackson. Anaphora resolution in the extraction of treatment history language from court opinions by partial parsing. In *Proc. of 7th ICAIL*, 1999.
3. Amit Bagga and Breck Baldwin. Algorithms for scoring coreference chains. In *The first international conference on language resources and evaluation workshop on linguistics coreference*, volume 1, pages 563–566. Granada, 1998.
4. Cristian Cardellino, Milagro Teruel, Laura Alonso Alemany, and Serena Villata. A low-cost, high-coverage legal named entity recognizer, classifier and linker. In *Proc. of 16th ICAIL*, 2017.
5. Cristian Cardellino, Milagro Teruel, Laura Alonso Alemany, and Serena Villata. Ontology population and alignment for the legal domain: Yago, wikipedia and lkif. In *Proc. of ISWC 2017*, 2017.
6. Joe Cheri and Pushpak Bhattacharyya. Coreference resolution to support ie from indian classical music forums. In *Proc. of RANLP*, pages 91–96, 2015.
7. C. Dozier and R. Haschart. Automatic extraction and linking of personal names in legal text. In *Proc. of Recherche dInformations Assistee par Ordinateur (RIAO-2000)*, 2000.
8. P. Jackson, K. Al-Kofahi, A. Tyrrell, and A. Vachher. Information extraction from case law and retrieval of prior cases. *Artificial Intelligence*, 150, 2003.
9. Sushanta Kumar, P. Krishna Reddy, V. Balakista Reddy, and Aditya Singh. Similarity analysis of legal judgments. In *Proc. of the COMPUTE'11*, 2011.
10. John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. of the Eighteenth International Conference on Machine Learning*, ICML '01, pages 282–289, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.
11. Xiaoqiang Luo. On coreference resolution performance metrics. In *Proc. of HLT-EMNLP*, pages 25–32, 2005.
12. Raquel Mochales and Marie-Francine Moens. Argumentation mining. *Artificial Intelligence and Law*, 19(1), 2011.
13. Vincent Ng. Machine learning for entity coreference resolution: A retrospective look at two decades of research. In *Proc. of the 31st AAAI Conference on Artificial Intelligence*, pages 4877–4884, 2017.
14. Haoruo Peng, Kai-Wei Chang, and Dan Roth. A Joint Framework for Coreference Resolution and Mention Head Detection. In *CoNLL 2015*, pages 12–21, 2015.
15. Haoruo Peng, Daniel Khashabi, and Dan Roth. Solving Hard Coreference Problems. In *NAACL HLT 2015*, pages 809–819, 2015.
16. S. Pradhan, X. Luo, M. Recasens, E. Hovy, V. Ng, and M. Strube. Scoring coreference partitions of predicted mentions: A reference implementation. In *Proc. of ACL*, 2014.

17. M. Saravanan, B. Ravindran, and S. Raman. Improving legal information retrieval using an ontological framework. *Artificial Intelligence and Law*, 17(2), 2011.
18. Olga Shulayeva, Advaith Siddharthan, and Adam Wyner. Recognizing cited facts and principles in legal judgements. *Artificial Intelligence and Law*, 25(1), 2017.
19. W. M. Soon, H. T. Ng, and D. C. Y. Lim. A machine learning approach to coreference resolution of noun phrases. *Computational linguistics*, 27(4), 2001.
20. Giulia Venturi. Semantic processing of legal texts. chapter Legal Language and Legal Knowledge Management Applications, pages 3–26. Springer-Verlag, 2010.
21. Marc Vilain, John Burger, John Aberdeen, Dennis Connolly, and Lynette Hirschman. A model-theoretic coreference scoring scheme. In *Proc. of the 6th conference on Message understanding*, pages 45–52, 1995.
22. Christopher Walker, Stephanie Strassel, Julie Medero, and Kazuaki Maeda. Ace 2005 multilingual training corpus. *Linguistic Data Consortium*, 57, 2006.
23. Mehdi Yousfi-Monod, Atefeh Farzindar, and Guy Lapalme. Supervised machine learning for summarizing legal documents. In Atefeh Farzindar and Vlado Kešelj, editors, *Proc. Canadian AI*. Springer, 2010.
24. P. Zhang and L. Koppaka. Semantics-based legal citation network. In *Proc. of the 11th ICAIL*, pages 123–130, 2007.