Incorporating Linguistic Expertise using ILP for Named Entity Recognition in Data Hungry Indian Languages

Anup Patel, Ganesh Ramakrishnan, Pushpak Bhattacharya

Department of Computer Science and Engineering, IIT Bombay, Mumbai – 400076, India
{anuppatel,ganesh,pb}@cse.iitb.ac.in

Abstract. Developing linguistically sound and data-compliant rules for named entity annotation is usually an intensive and time consuming process for any developer or linguist. In this work, we present the use of two Inductive Logic Programming (ILP) techniques to construct rules for extracting instances of various named entity classes thereby reducing the efforts of a linguist/developer. Using ILP for rule development not only reduces the amount of effort required but also provides an interactive framework wherein a linguist can incorporate his intuition about named entities such as in form of mode declarations for refinements (suitably exposed for ease of use by the linguist) and the background knowledge (in the form of linguistic resources). We have a small amount of tagged data - approximately 3884 sentences for Marathi and 22748 sentences in Hindi. The paucity of tagged data for Indian languages makes manual development of rules more challenging, However, the ability to fold in background knowledge and domain expertise in ILP techniques comes to our rescue and we have been able to develop rules that are mostly linguistically sound that yield results comparable to rules hand-crafted by linguists. The ILP approach has two advantages over the approach of hand-crafting all rules: (i) the development time reduces by a factor of 240 when ILP is used instead of involving a linguist for the entire rule development and (ii) the ILP technique has the computational edge that it has a complete and consistent view of all significant patterns in the data at the level of abstraction specified through the mode declarations. The point (ii) enables the discovery of rules that could be missed by the linguist and also makes it possible to scale the rule development to a larger training dataset. The rules thus developed could be optionally edited by linguistic experts and consolidated either (a) through default ordering (as in TILDE[1]) or (b) with an ordering induced using [2] or (c) by using the rules as features in a statistical graphical model such a conditional random field (CRF) [3]. We report results using WARMR [4] and TILDE to learn rules for named entities of Indian languages namely Hindi and Marathi.

Keywords: Named Entity Recognition, WARMR, TILDE, ILP

Presented at the 19th International Conference on Inductive Logic Programming (ILP'09).

1 Introduction

Identifying entities from unstructured text forms a very important part of information extraction systems. These entities are typically noun phrases and comprise of one to a few tokens in the unstructured text. Named entities like names of persons, locations, and companies are the most popular form of entities as popularized in the MUC [5][6], ACE [7][8], and CoNLL [9] competitions. Named entity recognition was first introduced in the sixth MUC [6] and consisted of three detection subtasks: proper names and acronyms of persons, locations, and organizations (ENAMEX), absolute temporal terms (TIMEX) and monetary and other numeric expressions (NUMEX). Early named entity recognition systems were rule-based with hand-crafted rules [10][11]. Since hand-crafting rules is tedious, algorithms for automatically learning rules were developed [12][13], but these approaches did not provide adequate mechanism for incorporating linguistic knowledge. This paper is organized as follows: Section 2 describes the complexity of Named Entity Recognition for Indian Languages, the motivation for using an ILP approach for this task and some specifics of the ILP approach. Section 3 we show our experimental results for the ILP and other approaches on Indian Language NER.

2 NER for Indian Languages using ILP

There has been a lot of work in NER for English and European Languages with claims for high precision and recall. The reason for success in these languages is a very rich tagged corpus and good linguistic insight about the usage of named entities. For Indian languages we do not have this privilege of huge tagged corpus which makes it difficult to have a good linguistic insight about named entities. The table below shows current status of tagged corpus for NER in Hindi and Marathi:

Language	Words	Person Tags	Organization Tags	Location Tags
Marathi	54340	3025	833	997
Hindi	547138	5253	2473	6041

Table 1. Hindi and Marathi named entity corpus

For further analyzing the efforts required for NER in Hindi and Marathi, we analyzed the tagged corpus and recorded some ambiguous cases which create problems in manually developing rules for named entities. Since both languages show similar ambiguities, we have listed some of ambiguities only for Marathi:

Ambiguity	Examples
Variations of Proper Nouns	• डॉ. काशिनाथ घाणेकर, डॉ. घाणेकर, डॉक्टर *(Dr. Kashinath Ghanekar, Dr. Ghanekar, Doctor)* • भारतीय जनता पार्टी, भा. ज. पा. *(Bhartiya Janta Party, B. J. P.)*
Person v/s	• डॉ. लागू/PER यांनी मनोगत मांडले

Adjective v/s Verb	*(Dr. Lagu expressed his thoughts)* • ही योजना संपूर्ण शहरात लागू/JJ करण्यात येणार आहे. *(This scheme will be applicable in the whole city.)* • ..... पण अजिबात झोप लागू/VM दिली नाही. *(..... but he didn't allow me fall asleep at all.)*
Person v/s Common Noun	• मुंबईला आल्यावर डॉक्टरांना/PER फोन करणे भागच होते. *(After coming to Mumbai it was must to call the Doctor.)* • तू डॉक्टर/NN की मी? *(Are you doctor or me?)*
Person v/s Organization	• नेताजींच्या/PER गूढ मृत्यूचा मागोवा ..... *(Following Netaji's suspicious death .....)* • "मिशन नेताजी/ORG' या स्वयंसेवी संस्थेने ..... *("Mission Netaji" is a voluntary organization that .....)*
Organization v/s Location	• पाक/ORG संघ/ORG शनिवारी लंडनमार्गे पाकला/LOC प्रयाण करणार आहे. *(The Pakistan team will go to Pakistan via London on Saturday)*
Person v/s Facility	• सरस्वती आणि लक्ष्मीची/PER एकत्रित उपासना केल्यास ..... *(If Saraswati and Laxmi are worshiped together .....)* • श्रीकृष्ण,सुंदर, लक्ष्मी/FAC अशी नाट्य मंदिरे होती. *(There were Drama Theaters like Shri Krishna, Sundar, Laxmi.)*
Location v/s Person	• निगडी येथील भक्ती शक्ती चौक, टिळक/LOC चौक/LOC, ... *(Bhakti Chauk, Tilak Chauk, ... from Nigdi)* • टिळक/PER व डॉ. बाबासाहेब आंबेडकर ..... *(Tilak and Dr. Ambedkar .....)*

Table 2. Ambiguities in named entities found in Indian languages
(**Note:** The abbreviations ORG=Organization, PER=Person, FAC=Facility, LOC=Location, NN=Noun, JJ=Adjective, and VM=Verb)

The above ambiguous cases motivate us to use ILP for learning named entity rules for Indian languages. Following are the benefits of using ILP for inducing named entity rules in Indian language:

i. **Incorporating linguistic expertise using mode declaration:** Developing hand-crafted rules for named entities in the presence of ambiguities could lead to rules that may produce false positives (in other words imprecise). This makes it difficult for a linguist to get a named entity rule correct in the first shot; (s)he has to undergo a number of iterations of manually refining each rule until the rule is precise enough. On the other hand, if the linguist used an ILP technique then (s)he needs to only give high-level specification for the search space of rules in the form of mode declaration for refinements of rules. The onus is then on the

ILP technique to produce rules with good confidence and/or support resulting in good overall precision. Our experience with NER for Hindi and Marathi shows that ILP techniques have a computational advantage in coming up with a good and consistent set of named entity rules in considerably less time compared to process of hand-crafting rules.

ii. **Incorporating linguistic expertise using background knowledge:** Since most of the Indian languages currently have very small tagged corpus, the linguist has to apply apriori knowledge about named entities while hand-crafting rules to cover cases not occurring in the tagged corpus. ILP techniques provide a principled approach of applying such apriori knowledge in the form of the background knowledge.

iii. **Covering all significant rules:** There is always a possibility of human error in covering all hand-crafted rules. Consequently a significant rule may be missed out. However, ILP techniques (such as WARMR) will never miss out any such rule that can be generated by successive application of mode declarations provided by the linguist. If the mode declarations are complete enough the ILP approach can yield all possible significant rules.

The above benefits illustrate that ILP does not substitute a linguist but it is excellently complements the linguist by helping him save efforts and also by improving his ability to come up with a complete set of significant rules. There are a number of ways in which we can use the rules learned by ILP, but for simplicity we suggest three ways of consolidating the learned rules in a named entity annotator:

a) Retain the default ordering of learned rules in the rule firing engine.
b) Induce an ordering on the learned rules using greedy heuristics such as [2].
c) Construct a feature corresponding to each rule, with the feature value 1 if the rule covers an instance and 0 otherwise. The features (which can be functions of both the head as well as the body of the rules) can be used in a statistical graphical model such as CRF [3].

We could use several ILP techniques for learning rules, but we shall experiment with only two techniques:

1. **WARMR:** This is an extension of the apriori algorithm to first-order logic. Typically apriori based techniques are computationally expensive. The resulting rules are not ordered and we need to explicitly induce ordering using some heuristic or greedy approach since ordering a decision list is a NP-hard problem [2]. Consolidation techniques **b)** and **c)** are suitable in this case.
2. **TILDE:** This is an extension of traditional C4.5 decision tree learner to first-order logic. Decision tree induction algorithms are usually greedy and hence computationally faster than WARMR like algorithms. Since a decision tree can be serialized to an equivalent ordered set of rules (decision list). Consolidation technique **a)** is suitable in this case.

3 Experimental Results

We have use a hand-crafted rule based named-entity recognizer for Marathi developed by a linguist using the GATE [14] system. The rules were hand-crafted over a period of 1 month (240 hours for 8 hours per day). We measured the performance of hand-crafted rule based system on a completely tagged corpus (3884 sentences and 54340 words).

We learnt Marathi named entity rules using the WARMR and TILDE systems available as a part of ACE [15] data mining system. For induction of rules using 80% (3195 sentences and 42635 words) of tagged corpus, TILDE took 1 hour and WARMER took 140 hours (5 days and 20 hours). This gives us and reduction in time for rule development by factor of 240 for TILDE and by a factor of 1.7 for WARMR. To compare the quality of the learnt rules we consolidated the rules and applied them over the remaining 20% (689 sentences and 11705 words) of the tagged corpus in following ways:

TILDE Rule Based NER: Rules learned by TILDE are plugged into a rule-based named entity recognizer without altering the order of rules.

WARMR Rule Based NER: Rules learned by WARMR are ordered using simple precision score heuristic and a greedy algorithm mentioned in [2]. These ordered rules are then plugged into a rule-based named entity recognizer.

WARMR CRF Based NER: Rules learned by WARMR plugged into CRF [16] as features ignoring the order of rules.

The performances of the hand-crafted rule based (HR), the TILDE rule based (TR), the WARMR rule based (WR), and the WARMR CRF based (WC) systems are shown below in Table 3 for Marathi.

Entity	Precision				Recall				F-Measure			
	HR	**TR**	**WR**	**WC**	**HR**	**TR**	**WR**	**WC**	**HR**	**TR**	**WR**	**WC**
PER	0.61	0.55	0.60	**0.74**	0.70	**0.99**	0.90	0.91	0.65	0.71	0.72	**0.82**
ORG	0.15	**0.85**	0.19	0.59	0.10	0.37	0.46	**0.52**	0.12	0.51	0.27	**0.55**
LOC	0.51	**0.54**	0.41	0.51	0.24	0.18	0.35	**0.45**	0.33	0.27	0.38	**0.48**

Table 3. Experimental results for Marathi

References

[1] H. Blockeel and L. D. Raedt, "Top-down induction of logical decision trees," in *Artificial Intelligence*, 1998.

[2] V. Chakravarthy, S. Joshi, G. Ramakrishnan, S. Godbole, and S. Balakrishnan, "Learning Decision Lists with Known Rules for Text Mining," in *The Third International Joint Conference on Natural Language Processing (IJCNLP 2008)*, 2008.

[3] J. Lafferty, A. McCallum, and F. Pereira, "Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data," in *Proceedings of the International Conference on Machine Learning (ICML-2001)*, 2001.

[4] L. D. a. L. D. Raedt, "Mining association rules in multiple relations," in *Proceedings of the 7th International Workshop on Inductive Logic Programming*, 1997, pp. 125-132.

[5] N. A. Chinchor, "Overview of MUC-7/MET-2," 1998.

[6] R. Grishman and B. Sundheim, "Message understanding conference-6: A brief history," in , 1996, p. 466–471.

[7] "Automatic content extraction (ACE) program," in *NIST*, 1998.

[8] "Annotation guidelines for entity detection and tracking," in *ACE*, 2004.

[9] E. F. Tjong Kim Sang and F. D. Meulder, "Introduction to the conll-2003 shared task: Language-," in *Seventh Conference on Natural Language Learning (CoNLL-03)*, 2003, p. 142–147.

[10] D. E. Appelt, J. R. Hobbs, J. Bear, D. J. Israel, and M. Tyson, "Fastus: A finite-state processor for information extraction from real-world text," in *IJCAI*, 1993, p. 1172–1178.

[11] E. Riloff, "Automatically constructing a dictionary for information extraction tasks," in *AAAI*, 1993, p. 811–816.

[12] M. E. Califf and R. J. Mooney, "Relational learning of pattern-match rules for information extraction," in *Proceedings of the Sixteenth National Conference on Artificial Intelligence (AAAI-99)*, 1999, p. 328–334.

[13] S. Soderland, "Learning information extraction rules for semi-structured and free text," in *Machine Learning*, 1999.

[14] H. Cunningham, D. Maynard, K. Bontcheva, and V. Tablan, "Gate: An architecture for development of robust HLT applications," in *Recent Advanced in Language Processing*, 2002, pp. 168-175.

[15] H. Blockeel and e. al. (2008, Mar.) Machine Learning Group - ACE Dataming System. [Online]. http://www.cs.kuleuven.be/~dtai/ACE/doc/ACEuser-1.2.12-r1.pdf

[16] S. Sarawagi. (2004) CRF Project Page. [Online]. http://crf.sourceforge.net/