IJREAT International Journal of Research in Engineering & Advanced Technology, Volume 1, Issue 2, April-May, 2013
ISSN: 2320 - 8791
www.ijreat.org

JOINT INFERENCE IN INFORMATION EXTRACTION USING MARKOV LOGIC

J. Refonaa
Computer Science and Engineering,
SSN College of Engineering, Anna University,
Chennai, TamilNadu, India

Abstract

Information extraction takes text or semi-structured data as the input and produces structured records as the output.Markov Logic is a Statistical Relational Learning technique used for learning from data which has relational struture. The goal of the project is to study the performance of a Statistical Learning Technique (SRL) such as Markov Logic for Information Extraction tasks. We use citation matching as a testbed for illustrating the use of Markov Logic for information extraction. Citation matching involves extracting bibliographic records from citation lists in technical papers and merging records that represent the same publication. We focus only on extracting titles, authors and venues from citation strings. We extract these fields, first by first segmenting each candidate record separately, and then merging records that refer to the same entities. A joint approach to information extraction is used where segmentation of all records and entity resolution are performed together in a single integrated inference process. We use Alchemy, a open source tool, for uaing Markov Logic and then intend to incorporate a different learning/inference algorithms for information extraction.

1. Introduction

Due to explosive growth of information in the present world, it is necessary that we need systems that automatically extract information from free-text without human support. This is one form of machine reading and it significantly reduces the human burden of manually extracting results. Machine learning methods are best suited for applications where:

- The application is too complex for people to manually design the algorithm.
- The application requires that the software customize to its operational environment.

Statistical relational learning extends statistical learning to relational representation of data. Markov Logic is a statistical relational learning technique; it is a probabilistic extension of first-order logic and suitable for handling complex data with uncertainty. When we have a large amount of unlabeled data and hence supervised learning is not feasible, statistical relational learning techniques such as Markov Logic facilitates unsupervised learning learning directly from unlabeled data.

2. RELATED WORK

2.1. Markov Logic in Information Extraction.

Markov Logic is a probabilistic extension of finite firstorder logic [1]. The complexity of objects in a domain can be represented using first-order logic and uncertainty can be handled using probability. It is a set of weighted first-order formulas. The weight of a feature is the weight of the firstorder formula that originated it. The probability of a state "x" in such a network is given by the log-linear model.

$$P(x) = \frac{1}{Z} \exp\left(\sum_i w_i n_i(x)\right)$$

where Z is a normalization constant, "wi" is the weight of the "i"th formula and "ni" is the number of satisfied groundings [1].

We apply Markov Logic that has a set of constraints and that follows some rules. Two positions of a citation are usually in the author field and the middle one in the title, initials (e.g., "J.") tend to appear in either the author or the venue field, positions preceding the last non-venue initial

are usually not part of the title or venue and positions after the first venue keyword.

It is used to extract the database records from the available sources. We first segment each record seperately and then merge the records that belong to the same entities. A joint approach where segmentation and entity resolution are performed together in a single integrated inference process. In the MLN, the predicate Token (t,i,c) is true iff token " t" occurs in position "i" of citation "c". Token can be a word, date, number, etc. Punctuation marks are not treated as separate tokens. HasPunc(c,i) returns true if the punctuation is present in position "i" in citation "c". Infield(i,f,c) returns true iff token in position "i" belongs to field "f " in citation "c".

2.2. Statistical Relational Learning

Statistical machine learning approaches are adept at handling noise and uncertainty in the data using probabilistic models. Logical approaches to learning take advantage of the structure in the data objects and the background knowledge by representing the data using first-order logic. Statistical relational learning (SRL) approaches combine the advantages of both first-order logic and probabilistic approaches. They can use first-order logic for representing complex and relational structure of data objects, they can also handle uncertainty with probabilistic methods.

Supervised learning require data to be labeled by a supervisor. Most of the learning techniques are supervised and use labeled data. But labeling data needs a lot of effort and is error prone. We have a lot of unlabeled data. Unsupervised learning aims at learning from unlabeled data, and it is more challenging than supervised learning. Statistical relational learning techniques are particularly suited for unsupervised learning since they can take advantage of the relation among the structured data objects and use joint inference for learning in place of labels on the data.

3. METHOD

3.1. Isolated Segmentation.

Isolated segmentation model is essentially a Hidden Markov Model(HMM). It is used to detect field boundaries. We use observation matrix that correlates tokens with filelds and is represented by the simple rule.

Token (+t, i, c)) => InField (i, +f, c)

If the token is present in position "i" in citation "c", then it implies that it belongs to the particular field "f". The transition matrix of the HMM is represented by a rule of the form.

InField (i, +f, c)) => InField (i + 1, + f′, c)

If token " i " belongs to field "f " and if "i+1"th token belongs to a different field, then the f is replaced by f '. Position i+1 is used to check whether the token is present in the same field or different filed.

InField (i, + f, c) ^¬ HasPunc (c, i))=>
InField (i + 1, +f, c)

The clear boundary between the two fields is identified by using the punctuation. By using HasPunc (c,i), we separate the tokens that are present in different fields.

3.2. Entity Resolution.

Entity Resolution is used to identify duplicate records. We follow a set of rules to check where the citations have common tokens and if fields too match then the two citations also match with each other.

SimilarTitle (c, i, j, c′, i′, j′) ^ SimilarVenue(c, c′)
=> SameCitation(c, c′)

To verify whether two citations have similar title or similar venue, the predicates *SimilarTitle(c, i , j, c′, i′, j′) and SimilarVenue(c, c′)* are used. If the token in position [i, j] in citation c is similar to the token in position [i′, j′] in citation c′, then the two citations are similar.

3.3. Joint Segmentation

Segmenting a citation can help segment similar ones. This idea leads to joint segmentation. One citation do not have punctuation whereas the other has a clear boundary between the fields. To obtain the fields of former citation, the latter citation boundary is used. This can be achieved by using JointInferenceCandidate (c ,i , c).

InField (i, +f, c) ^ ¬ HasPunc (c, i)
^ (¬∃c JointInferenceCandidate (c, i, c′))
=> InField (i + 1, +f, c)

The trigrams from the position i from c are compared to those in c'. Consider the two citations

- R. Schapire. On the strength of weak learnability. Proceedings of the 30th I.E.E.E. Symposium on

the Foundations of Computer Science, 1989, pp. 28-33.

- Robert E. Schapire. 5(2) The strength of weak learnability. Machine Learning, 1990 197-227.

The second citation has a clear boundary which can be used to sement the first citation. Consider the trigram "the strength of". If the first citation is also segmented using the same trigram, then the author field becomes "R. Schapire. On". To overcome this disadvantage the following rule is used.

InField (i, +f, c) ^ ¬ HasPunc(c, i)
^ (¬∃cJointInferenceCandidate(c, i, c′)
^ SameCitation(c, c′)) => InField(i + 1, +f, c).

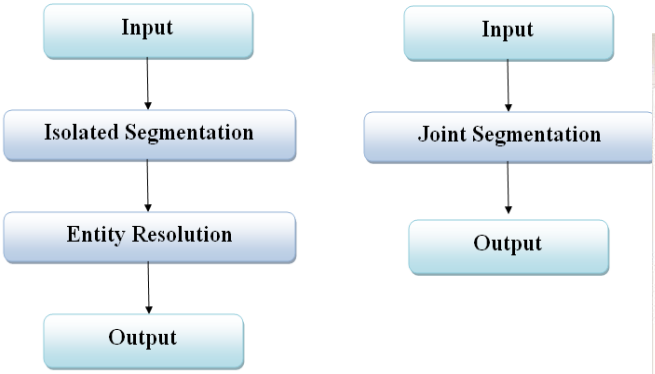


Fig. 1: Citation Matching Architecture Diagram

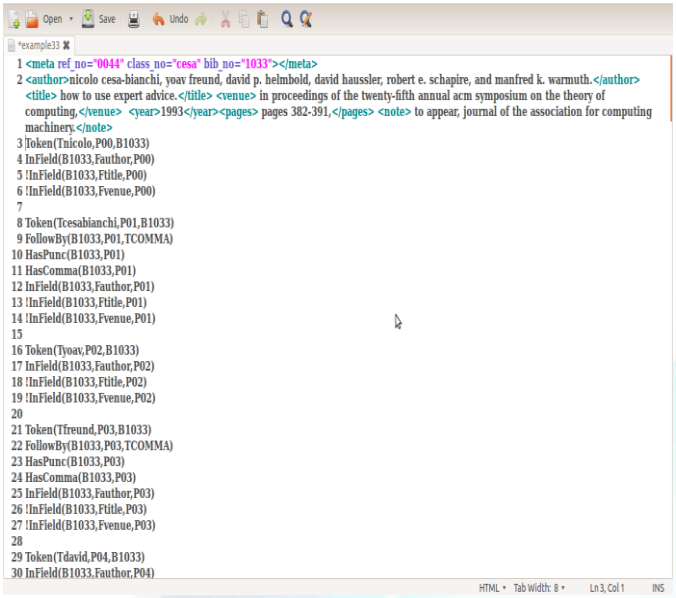
3.4. Tool — Alchemy

Alchemy is an open source software and is used for systematically processing the Markov Logic theories. Alchemy is used for the purpose of statistical relational learning in general and used for Markov Logic, in particular. Alchemy performs three basic tasks namely structure learning, weight learning and inference. Strucure learning checks whether the extracted sentences are in the proper order. Weight learning is used to assign weights to the frequently accessed word. Inference is used to check to what extent these trained ground atoms and the input query atoms are similar. In citation extraction data considered, the ground atoms are (author, title,venue) and they are formed as relations. From these relations multiple relations are formed. The ground atom is trained using structure learning, weight learning and inference. The query ground atom and the trained ground atom will be compared. During comparison stage statistical relational learning and Markov Logic is used. Here text classifier is used to classify the author, title,venue.

4. EXPERIMENTAL RESULTS

MC-SAT is a slice sampling Markov Chain Monte Carlo (MCMC) algorithm. It uses a combination of satisfiability testing and simulated annealing to sample from the slice. MCSAT is an order of magnitude faster than previous MCMC algorithms like Gibbs sampling and simulated tempering, and makes efficient joint inference. The voted perceptron algorithm optimizes weights by doing gradient descent on the conditional log-likelihood of the query atoms, given the evidence ones. The gradient with respect to a weight is the difference between the number of times the corresponding clause is true in the data and its expectation according to the MLN. MC-SAT tends to give more reliable results.

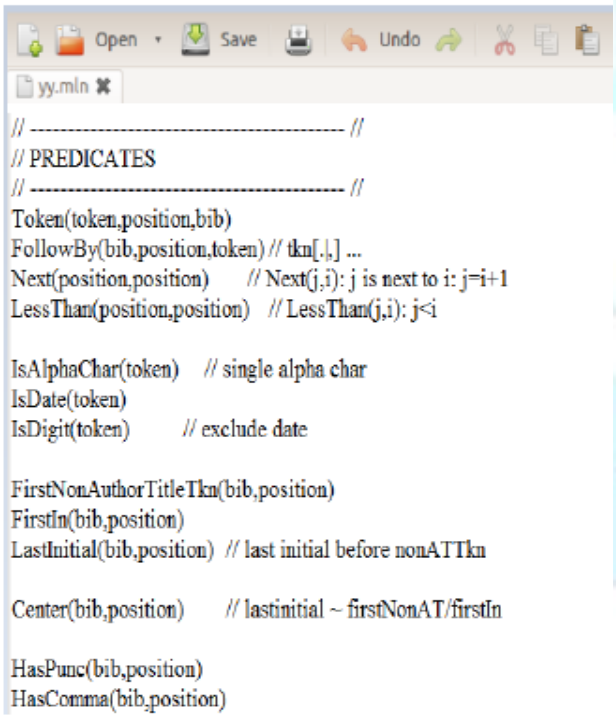
Fig. 2: citations



```
1 <meta ref_no="0044" class_no="cesa" bib_no="1033"></meta>
2 <author>nicolo cesa-bianchi, yoav freund, david p. helmbold, david haussler, robert e. schapire, and manfred k. warmuth.</author>
  <title> how to use expert advice.</title> <venue> in proceedings of the twenty-fifth annual acm symposium on the theory of
  computing,</venue>  <year>1993</year><pages> pages 382-391,</pages> <note> to appear, journal of the association for computing
  machinery.</note>
3 Token(Tnicolo,P00,B1033)
4 InField(B1033,Fauthor,P00)
5 !InField(B1033,Ftitle,P00)
6 !InField(B1033,Fvenue,P00)
7
8 Token(Tcesabianchi,P01,B1033)
9 FollowBy(B1033,P01,TCOMMA)
10 HasPunc(B1033,P01)
11 HasComma(B1033,P01)
12 InField(B1033,Fauthor,P01)
13 !InField(B1033,Ftitle,P01)
14 !InField(B1033,Fvenue,P01)
15
16 Token(Tyoav,P02,B1033)
17 InField(B1033,Fauthor,P02)
18 !InField(B1033,Ftitle,P02)
19 !InField(B1033,Fvenue,P02)
20
21 Token(Tfreund,P03,B1033)
22 FollowBy(B1033,P03,TCOMMA)
23 HasPunc(B1033,P03)
24 HasComma(B1033,P03)
25 InField(B1033,Fauthor,P03)
26 !InField(B1033,Ftitle,P03)
27 !InField(B1033,Fvenue,P03)
28
29 Token(Tdavid,P04,B1033)
30 InField(B1033,Fauthor,P04)
```

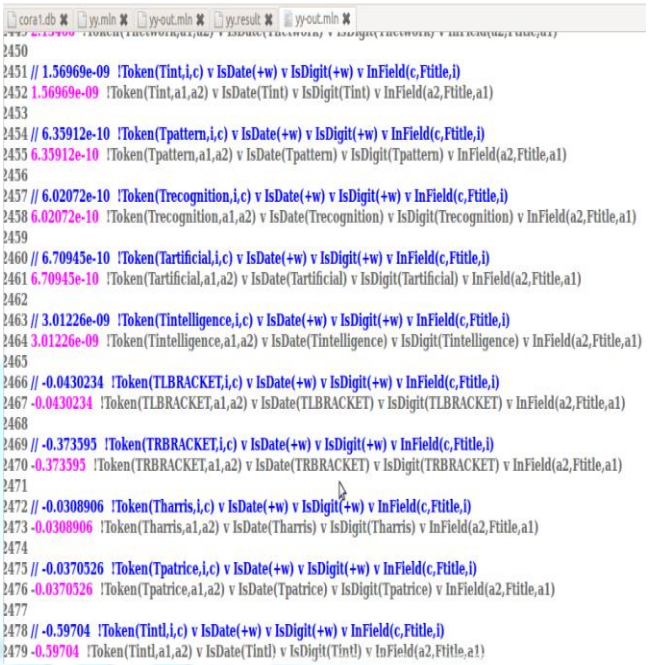
Fig. 3: Citations in Alchemy Format

4.1. Learning Weights



```
// ------------------------------------------- //
// PREDICATES
// ------------------------------------------- //
Token(token,position,bib)
FollowBy(bib,position,token) // tkn[.,] ...
Next(position,position)      // Next(j,i): j is next to i: j=i+1
LessThan(position,position)  // LessThan(j,i): j<i

IsAlphaChar(token)    // single alpha char
IsDate(token)
IsDigit(token)        // exclude date

FirstNonAuthorTitleTkn(bib,position)
FirstIn(bib,position)
LastInitial(bib,position)  // last initial before nonATTkn

Center(bib,position)       // lastinitial ~ firstNonAT/firstIn

HasPunc(bib,position)
HasComma(bib,position)
```

Fig. 4: Input sample



```
2450
2451 // 1.56969e-09  !Token(Tint,i,c) v IsDate(+w) v IsDigit(+w) v InField(c,Ftitle,i)
2452 1.56969e-09  !Token(Tint,a1,a2) v IsDate(Tint) v IsDigit(Tint) v InField(a2,Ftitle,a1)
2453
2454 // 6.35912e-10  !Token(Tpattern,i,c) v IsDate(+w) v IsDigit(+w) v InField(c,Ftitle,i)
2455 6.35912e-10  !Token(Tpattern,a1,a2) v IsDate(Tpattern) v IsDigit(Tpattern) v InField(a2,Ftitle,a1)
2456
2457 // 6.02072e-10  !Token(Trecognition,i,c) v IsDate(+w) v IsDigit(+w) v InField(c,Ftitle,i)
2458 6.02072e-10  !Token(Trecognition,a1,a2) v IsDate(Trecognition) v IsDigit(Trecognition) v InField(a2,Ftitle,a1)
2459
2460 // 6.70945e-10  !Token(Tartificial,i,c) v IsDate(+w) v IsDigit(+w) v InField(c,Ftitle,i)
2461 6.70945e-10  !Token(Tartificial,a1,a2) v IsDate(Tartificial) v IsDigit(Tartificial) v InField(a2,Ftitle,a1)
2462
2463 // 3.01226e-09  !Token(Tintelligence,i,c) v IsDate(+w) v IsDigit(+w) v InField(c,Ftitle,i)
2464 3.01226e-09  !Token(Tintelligence,a1,a2) v IsDate(Tintelligence) v IsDigit(Tintelligence) v InField(a2,Ftitle,a1)
2465
2466 // -0.0430234  !Token(TLBRACKET,i,c) v IsDate(+w) v IsDigit(+w) v InField(c,Ftitle,i)
2467 -0.0430234  !Token(TLBRACKET,a1,a2) v IsDate(TLBRACKET) v IsDigit(TLBRACKET) v InField(a2,Ftitle,a1)
2468
2469 // -0.373595  !Token(TRBRACKET,i,c) v IsDate(+w) v IsDigit(+w) v InField(c,Ftitle,i)
2470 -0.373595  !Token(TRBRACKET,a1,a2) v IsDate(TRBRACKET) v IsDigit(TRBRACKET) v InField(a2,Ftitle,a1)
2471
2472 // -0.0308906  !Token(Tharris,i,c) v IsDate(+w) v IsDigit(+w) v InField(c,Ftitle,i)
2473 -0.0308906  !Token(Tharris,a1,a2) v IsDate(Tharris) v IsDigit(Tharris) v InField(a2,Ftitle,a1)
2474
2475 // -0.0370526  !Token(Tpatrice,i,c) v IsDate(+w) v IsDigit(+w) v InField(c,Ftitle,i)
2476 -0.0370526  !Token(Tpatrice,a1,a2) v IsDate(Tpatrice) v IsDigit(Tpatrice) v InField(a2,Ftitle,a1)
2477
2478 // -0.59704  !Token(Tintl,i,c) v IsDate(+w) v IsDigit(+w) v InField(c,Ftitle,i)
2479 -0.59704  !Token(Tintl,a1,a2) v IsDate(Tintl) v IsDigit(Tintl) v InField(a2,Ftitle,a1)
```

Fig. 5: Output

4.2. Java API

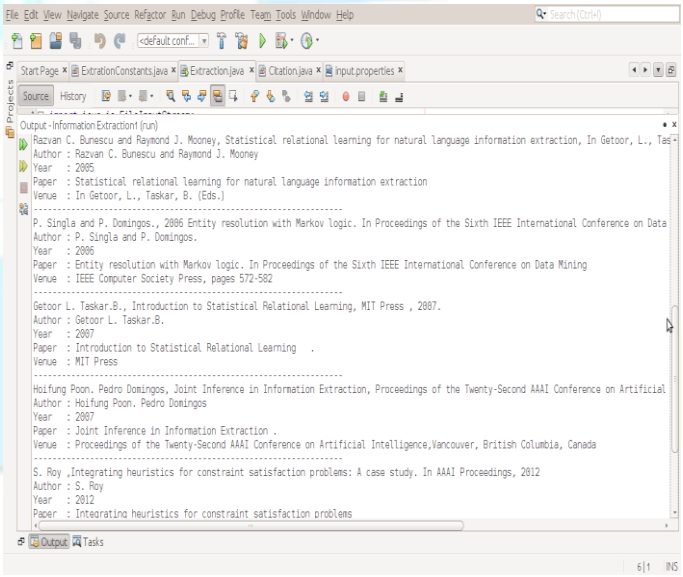


Fig. 6: Output

5. CONCLUSION

In this paper we show how the problem of un certainty with complex data can be addressed in citation matching domain using Markov logic, MC-SAT algorithm. Isolated segmentation module provides the results for weight learning and compares it with the other modules for the effective performance. The entity resolution module avoids duplicates and provides accuracy to the output result. Experiments are conducted on various dataset and accurate weights are obtained for citation list. The future work, is to implement these modules using the Java API.

6. ACKNOWLEDGMENT

The authors would like to thank the reviewers for their valuable comments that helped to improve this paper.

7.REFERENCES

[1] Hoifung Poon, Pedro Domingos, Markov Logic, 2011.
[2] Hoifung Poon, Pedro Domingos, Joint Inference in Information Extraction, Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence, July 22-26, 2007, Vancouver, British Columbia, Canada.
[3] Razvan C. Bunescu and Raymond J. Mooney, Statistical relational learning for natural language information extraction, In Getoor, L., Taskar, B. (Eds.), 2005.
[4] P. Singla and P. Domingos. Entity resolution with Markov logic. In Proceedings of the Sixth IEEE International Conference on Data Mining, IEEE Computer Society Press, pages 572-582, Hong Kong, 2006.
[5] Getoor, L., Taskar, B., eds, Introduction to Statistical Relational Learning, MIT Press , 2007.
[6] Wellner, B.; McCallum, A., Peng, F., Hay, M., An integrated, conditional model of information extraction and coreference with application to citation matching, 2004.
[7] Richardson, M. Domingos, P., Markov logic networks.Machine Learning pages 107-136, 2006.