Semi-supervised learning framework for knowledge extraction in Cricket domain

C C M T Fernando,W A V S Cooray,T G H Indeewara, D S K Jayasinghe, Subha Fernando
Faculty of Information Technology,University of Moratuwa
Katubedda, Sri Lanka

Abstract

Today's web is overwhelmed by the data and it is continuously growing, therefore processes of information retrieval and analysis have become tedious tasks. Although many machine learning approaches have been applied to mine these data, many of them are hardly succeeded in their approaches because they have restricted themselves into targeted or downloaded databases. Growing web can not simply be classified or mined by using static knowledge base, system has to grow with the web. Therefore, a system that can mine while learning from the mined data, is required. This paper proposes a framework that acts as a learning model to derive information by building relationships between different entities in online content by relying on few seeds being fed to the system at the start. Couple of extractors are used to derive facts based on their mutual correlations. Those facts have been occupied an ontology to generate new relationships and entities as candidates. A query system has been embedded to the miner to enable querying the knowledge base to retrieve appropriate outputs corresponding to a particular query. The system has evaluated against the cricket online sources.

Key Terms:

Machine Learning, Natural Language Processing, Self-Learning Machines.

1. INTRODUCTION

Machine learning is a highly promising area of AI and its applications are now being used in various fields starting from simple classification problems to critical business analysis scenarios. Traditional machine learning techniques such as supervised leaning, unsupervised learning and topic modeling have been the spine of those applications and despite the success they were majorly specified to perform a one given classification problem and require a vast amount of resources as they are trained with large amount of labeled examples. Semi-supervised learning approach defines learning from a handful set of seed examples and a large text corpus. NELL (Never Ending Language Learning) [1] is a program developed by CMU which exploits semi-supervised learning paradigm. NELL has been trained to learn from web and expand its ontology simultaneously using semi-supervised learning methods. This theses covers exploiting these learning approaches to learn information belong to a particular topic, in here, Sports category and use the learned facts to expand existing knowledge.

The semi-supervised learning problem is defined more likely to be humanistic other than fully supervised mechanism. The system is supposed to learn from reading and expand the knowledge with previously learned facts given a set of seed examples and a large text corpus. The general approach to create a semi-supervised learner is based on Never Ending Language Learning paradigm. We present that the Never Ending Learning can be extended to learn a particular domain and make the knowledge available to general public via more natural language related manner.

2. RELATED WORK

Though the area of semi-supervised learning has been come into the appearance recently, a plenty of researchers have done their researches in the domain. Instead of supervised learning classifiers, using of semi-supervised learning classifiers have been introduced exploiting the advantage of ability to depend on a small number of training data. Extracting instances and relationships from various forms of texts is performed by learning different extractors. Carlson et al.2010[2] introduces a set of semi-supervised learning information extractors that can run on freeform and semi-structured texts and extract instances of categorical data and relationship instances between categories. Also, a learner that can learn from morphological patterns of text were introduced in that research with constraints to be enforced to the learners.

Never ending language learning is a concept that creates a continuously growing ontology. The proposed architecture consists of subsystem components, candidate facts, beliefs and knowledge integrator to promote candidate instances and patterns. Later,OpenEval based techniques for CPL [3] were introduced while entity resolution (Function that classify noun phrases by they are being synonyms or not). Moving further, to overcome the problem of manually defining categories and relationships in semi-supervised learning approaches. Mohamed et al., 2011 [4] introduces an approach to automatic discovering of relationships between categories which later exploited to automatic ontology expansion.

3. APPROACH

The architecture for the learning system we have used has been introduced in NELL [5]. The prototype implementation of the project learns two types of knowledge.

(1) Instances of different categories of the sport: (players, venues, bowling style etc.)
(2) Semantic relations that couple the instances: (_ is a batsman/ground/bowler etc.)

The knowledge will be extracted by learning a freeform text extractor and a semi structured web text learner namely coupled pattern learner (CPL) and coupled SEAL (CSEAL) [2]. Nave Bayes text classification is used to classify the extracted web articles prior to information extraction to make sure articles are in the appropriate domain of sports. The semi-supervised learners are provided with an initial ontology as seed instances and relationships to start the mining of facts.

The derived instances will then be sent to knowledge integration sub component to check for the coupling constraints

and they will be promoted to the knowledge base unless they violate the constraints. Then the knowledge base will be exposed via a SPARQL endpoint to a natural language interface.

3.1 Knowledge extraction sub-component

The knowledge extraction sub-component is responsible for explore through the internet, extract sports related web content and then extract new knowledge based on recently upgraded patterns. Here we have used Crawler4J [6] implementation and initialized the crawler with seed URLs related to Sports. The HTML content is then parsed to extract text contents for CPL and the raw HTML content was used for CSEAL. Each run of the web crawler was limited to 100 web pages to achieve the efficiency while parsing the text corpus which is generated during each run. Each iteration returns with a raw text corpus of approximately 150,000-300,000 words. To remove unwanted characters and noisy content such as content that directly converted to plain text from tables and figures, preprocessing has been performed, r.f. fig.1.

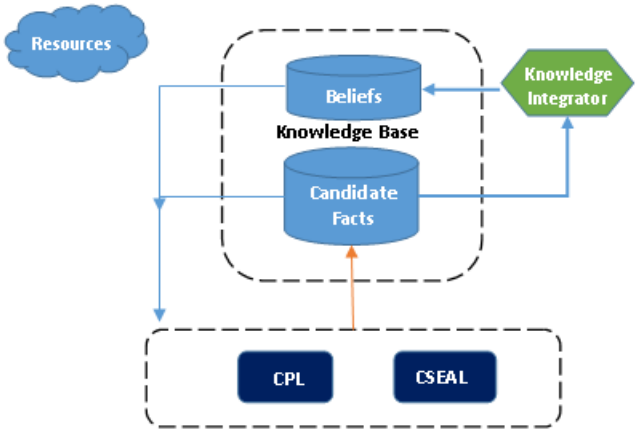

Figure 1: High Level Architecture of the System

3.2 Classification of web content

Considering the large amount of outgoing URLs in sports web pages related to commercial advertising and unrelated topics, the need for a text classifier was emerged. To overcome the problem, we used Nave Bayes classifier trained with labeled positive and negative data to classify the incoming web content. One of the problems that we had to deal with was the huge amount of features and the noisy content in the articles to be classified. Therefore, the feature reduction process was performed using Chi-square feature extraction process [7]. To select the significant features from the articles, we set the null hypothesis:

H_0 : The feature and sport(here it is cricket) are independent.

This allowed us to apply the following model to extract the features:

Let $N_{ec} = 1$ indicates the number of documents belongs to the class cricket and $N_{ec} = 0$ be the number of documents does not belong to the class cricket. Number of documents contain the particular feature term and the number of documents that does not contain the considered feature term defines by $N_{et} = 1$ and $N_{et} = 0$. Expectation for a given event to be happening is represented by an event, E_{event}.

Table 1: CALCULATING χ^2 VALUE TO DETERMINE THE CLASS OF A FEATURE

	$N_{ec} = 1$	$N_{ec} = 0$	
$N_{et} = 1$	67	11	78
$N_{et} = 0$	9	75	84
	76	86	162

For example, Let consider the class as ”Cricket” and the feature as ”team”. Then we can apply the Chi-square feature extraction under the null hypothesis as shown in table.1.:

Applying the data in table.1 in eq.1 give the value for χ^2_n as 92.01, where$n$ is the degrees of freedom, here $n = 1$

$$\chi^2 = \sum_{c\in\{0,1\}} \sum_{t\in\{0,1\}} \frac{(N_{e_te_c} - E_{e_te_c})^2}{E_{e_te_c}} \quad (1)$$

Given n =1 and by taking the level of significance (α) as = 0.01, the critical value for χ^2_1 is 6.63. Since the calculated χ^2_1 value ¿ 6.63, the H_0 is not rejected.

Given n is 1 and the probability to abandon the null hypothesis with 0.01, we use the critical χ^2 value 6.63. Since χ^2 value $>$ 6.63, the term ”team” will be picked up as a feature since the null hypothesis is rejected.

3.3 Learning of the facts

The fundamental task of the system, which is to learn facts from online resources, is performed by a learning subsystem. The learning system is comprised of two learners known as Coupled Pattern Learner (CPL) and Coupled SEAL learner (CSEAL). CPL is an unstructured text learner that makes use of semantic patterns of sentences while CSEAL extracts knowledge from semi structured text formats found in the web such as lists and tables.

The CPL is initialized with a starting ontology of category instances (instances for Players, Batsmen, Bowler etc.) and a set of text extracting patterns as seed inputs. Each iteration of CPL is ran for 2 passes, one for extracting new instances using the given patterns and extracting new patterns from recently promoted instances. For the first pass, the text corpus generated by the feature extraction component is used and searched for occurrences of promoted patterns despite the adjectives and determiners. For example, arg_1 *is a bowler*

The value of arg_1 will then be categorized an instance for the category the pattern is belonged to. For the next pass, each sentence of the text corpus is searched for occurrences of recently promoted category instances and referenced sentences are POS tagged in order to retrieve new text mining patterns with following sequences. Note that the sequences were annotated with *Penn English tree bank annotations*.

To extract patterns following a category instance, the structure shown in table.2. was used. Similarly, to extract patterns preceded to a category instance, the structure shown in table.3 or table.4 were used.

Those sequences are recognized by regular expressions to achieve higher efficiency. Once a candidate instance or candidate pattern is extracted, they will be ”type” checked to

Table 2: EXTRACTING PATTERNS FOLLOWING A CATEGORY INSTANCE

Category Instance	Possession	Adjective	Verb	Noun-phrase	Preposition
{1,2}	{0,1}	{0,*}	{1}	{0,1}	{0,1}
NNP	POS	JJ JJR JJS	VB VBD VBG VBN VBP VBZ	NN NNP NNS NNPS	IN

Table 3: EXTRACTING PATTERNS PRECEDED TO A CATEGORY INSTANCE - OPTION 1

Noun-phrases	Verbs	Adjective	Preposition	Determiners	Category Instance
{0,*}	{1}	{0,*}	{0,1}	{0,1}	{1,2}
NN NNP NNS NNPS	VB VBD VBG VBN VBP VBZ	JJ JJR JJS	IN	DT	NNP

make sure they belong to the right category. For a candidate instance, if it is referenced three times more than the number of times it referred with a mutually exclusive category, and then it is considered to be belonged to the particular category. Candidate patterns will also be filtered under the same constraint.

Seed relationships are stored for initial iterations with placeholders for given category instances. For example:

arg_1 *is a/an* arg_2 *Cricketer* :- matches to the sentence, *Ishant Sharma is an Indian Cricketer*, the this will populates the relationship: *Ishant Sharma is A MemberofTeam Indian*.

If both arguments of promoted relationship instances are found within sentences of the text corpus, then in-between sequence of words are extracted as a candidate relationship pattern for that relationship.

3.4 Natural Language Query Interface

Search the ontology to filter the features and patterns, to recall the instances, etc is a key aspect of this research since it allows the end user to browse the ontology via natural language queries. This is done in the form of a web interface which is easily available on all types of devices. Since the text interface is familiar to end users, we have implemented the searching of the ontology using natural language query processing.

Therefore the work discussed in this paper is focused on providing access to information which is stored in an ontology through natural language queries. As discussed in the above, the scope of this knowledge base is limited to *cricket*.

Table 4: EXTRACTING PATTERNS PRECEDED TO A CATEGORY INSTANCE - OPTION 2

Adjective	Noun-phrases	Adjective	Preposition	Determiners	Category Instance
{0,*}	{1,*}	{0,*}	{0,1}	{0,1}	{1,2}
JJ JJR JJS	NN NNP NNS NNPS	JJ JJR JJS	IN	DT	NNP

The querying of the ontology requires the use of some formal languages such as SPARQL or SeRQL. Though using formal languages to type queries provides a high level of control and expressiveness, it lacks the user-friendly interfaces.

There are different approaches to implement user-friendly knowledge base accessing systems. A graphical interface that allows users to browse the ontology, forms-based interface to perform semantic search based on ontology while encapsulating the complexity of formal languages, etc are some of those techniques. Here in this research, we have used a simple text box that takes Natural Language queries as an input to browse the ontological data[8].

3.5 The question-answering system

This system works by converting natural language query input to a formal semantic query. We use Jena framework to implement the ontology, in which the converted query will be searched in terms of SPARQL query terms because the system is configured to generate this formal query language.

The system is robust, that it needs to attempt to identify the input it gets without relying on syntax, grammatically correct queries or the level of context to check disambiguation through linguistic analysis. In our approach, we focus more on leveraging the information encoded in the ontology and use lightweight linguistic processing of the text input query. When a query is received, all of the words may not match ontology concepts, instead there is an unmatched part of textual query that can be used to predict property names for disambiguation. The sentence is then used to identify the structure of the query where a parse tree is generated through Natural Language parser. The property names is converted into a formal query so that it can be executed against the ontology. In this process until the answer is generated, a series of techniques is used to identify the possible interpretations which allows filtering low scoring options that will reduce the ambiguity.

The developed ontology is based on Jena framework and it stores data in the owl-rdf format. Therefore to implement the query system we use jOWL which is a jquery plugin for navigating and visualizing owl-rdfs documents. The front end of the query system is developed using jsp and servlets.

3.6 Initialization of the system

We have selected a specific set of queries in the application, such as:
- Who is $< someone >$, e.g. Who is sanga?
- What is $< something >$, e.g. what is the highest test score of Sri Lanka?

Apache OpenNLP[9] is used as a toolkit for the processing of NL query. To generate the parse tree we use part-of-speech tagging and parsing techniques embedded in the library. We define regular expressions, $regex$ that can match natural language questions and translate them into an abstract semantic representation using python. This will define which questions the QA system can handle and what to do with them.

The regular expression for the input question can be written as follows:
$regex$= *Lemma("what") + Lemma("be") + target + Question(Pos("."))*

The interpret method is called when a $regex$ has successfully found a match to an input question, which specifies the semantics of the matched question:

```
Laxman  | Batsman | argument scored
Gautam Gambhir  | Batsman | argument scored
Laxman  | Batsman | argument scored
Laxman  | Batsman | argument scored
Laxman  | Batsman | argument scored
Brendon McCullum  | Batsman | argument scored
Tendulkar  | Batsman | argument has scored
Sangakkara  | Batsman | argument has scored
Sangakkara  | Batsman | argument became the second highest run scorer
Dhoni  | Batsman | argument averages
Welsh  | Batsman | argument averages
Ponting  | Batsman | argument scored his
Dhoni  | Batsman | argument scored his
Harbhajan  | Batsman | argument scored his
```

Figure 2: Extraction of instances from the available relations

```
"C:\Program ...
Sentence : Ishant Sharma is an Indian Cricketer.
argument 1 : Ishant Sharma , argument 2 : Indian
Predicate : IsAMemberOfTeam(Ishant Sharma, Indian)

Sentence : Kumar Sangakkara is a Sri Lankan Cricketer.
argument 1 : Kumar Sangakkara , argument 2 : Sri Lankan
Predicate : IsAMemberOfTeam(Kumar Sangakkara, Sri Lankan)

Sentence : Ponting is an Aussie Cricketer.
argument 1 : Ponting , argument 2 : Aussie
Predicate : IsAMemberOfTeam(Ponting, Aussie)

Process finished with exit code 0
```

Figure 3: Extraction of relations from the available instances

```
def interpret(self,match):
        thing = match.target.tokens
        target = HasKeyword(thing)
        definition = IsDefinedIn(target)
return definition
```

The query generated for the Natural Language question 'who is sanga?' will be as follows[10].

```
SELECT DISTINCT ?x_1 WHERE :
        ?x_0 rdf:typefoaf:Person.
        ?x_0 rdfs:label "sanga"@en.
        ?x_0 rdfs:comment ?x_1.
return definition
```

4. EVALUATION

For the first iteration of the system, web crawling subsystem has run to extract information from online sources. It was executed with a given set of seed URLs and with search depth of 5 and a limit of browsing maximum 100 web pages. The result text corpus is then subjected to thorough filtering process to eliminate noisy content such as non-English texts, copyright texts, remove referencing, to add spaces between sentences and to remove sentences without a verb and sentences that have repeating characters and total uppercase letters. All the filtering was performed using regular expressions.

That reduced the size of the text corpus from 240,663 words to 52,699 words but with meaningful 3125 sentences. Few categories were initialized with seed instances and a few promoted patterns were added to each category. Then the first pass of the CPL was ran to extract new candidate patterns from given seed instances and some of the resulted patterns are showed in table.5. Then each seed pattern was searched in the text corpus and the extracted candidate instances as shown on table.6.

Table 5: EXTRACTING PATTERNS FROM THE SEED-INSTANCES

Some candidate patterns	Category	Total extracted patterns	Patterns that are reoccurred
argument to the finals of the World Cup, argument went on to win both matches by, argument salvage a draw and, argument Under–19s in, argument has won 18	Team	393	argument v India, argument for a two test series with, argument captain
argument officially announced his retirement from, argument was a key member of the team, argument scored 2868 runs in year, argument has 93 half-centuries	Batsman	495	argument has 38 centuries , argument has scored, argument scored his
argument took 24 wickets at, argument returned to take the new ball, argument once bowled 152 kph	Bowler	116	argument scalps five, argument managed 11 wickets, argument had a bowling average

Table 6: EXTRACTING INSTANCES FROM THE SEED-PATTERNS

Promoted pattern	Category	Some extracted instances	Accuracy
argument in the	Team	India U19, Mumbai Indians, Odisha,Test, Lancashire, England, Chennai Super Kings, West Zone, South, Ishant, WasimAkram, England's, Australia, South Zone, Rahul, Debut, Ludhiana, Sunrisers Hyderabad, Pakistan	56.50%
argument scored	Batsman	Sangakkara, Laxman, GautamGambhir, Brendon McCullum, Pietersen, Anwar, Dhoni , Rohit Sharma, Harbhajan, SanathJayasuriya	85.18%
argument took $[a-zA-Z\_0-9]\{1,\}$ wickets	Bowler	Lee, Harbhajan, Jadeja	100%

5. DISCUSSION

This research proposed a method that can be used to extract categories, relations or pattern instances. The method is capable of learning itself by updating its own ontology structure and instance base. The system learns facts by extracting instances from the pattern of seeds we provided. Gradually from these learn-instances it extends the ontological framework to identify more complicated patterns related to the given concept. This process is an iterative process, which never ends: i.e. extract instances from the patterns or categories, then extract complicated or hidden patterns from the browsed instances, and so on. Furthermore, the system is not only capable of expanding its knowledge bases but also semantically answer queries that raised by the users. This has been featured by the enrich ontological databases that it has learn in its journey. Moreover, the significant feature of this NLP queries is that user has the ability to raise the queries in human language.

The proposed system is capable of answering questions related only to the cricket domain, the future direction of this research would be expanding it to other domains, while

identifying the techniques required to remove the semantic conflicts between domains, and introduce the capability of the inter-operatability which enable it to read from any sources in the Web irrespective to the language it has written.

References

[1] Mitchell, T. M., Cohen, W., Hruschka Jr., E. R., Talukdar,P.,Betteridge,J., Carlson,A.,Dalvi,B., Gardner,M.,Kisiel,B.,and Krishnamurthy,J. Never-Ending Learning.

[2] Carlson, A., Betteridge, J., Wang, R. C., Hruschka Jr., E. R. and Mitchell, T. M. 2010b. Coupled semi-supervised learning for information extraction.

[3] Etzioni, Oren, et. al. 2011.Open information extraction.

[4] Mohamed, T. Hruschka Jr., E. R. and Mitchell, T. M. 2011. Discovering relations between noun categories.

[5] Carlson, A., Betteridge, J., Kisiel, B., Settles, B., HruschkaJr, E. R., and Mitchell, T. M. 2010a. Toward an architecture for never-ending language learning.

[6] Java Implementation of Crawler4J.[online] Available : https://code.google.com/p/crawler4j/

[7] Manning, Christopher D., Raghavan, Prabhakar, Schtze, Hinrich, Text classification and Naive Bayesin book - Introduction to Information Retrieval.

[8] Tablan,Valentin,Damljanovic, Danica, Bontcheva, Kalina, A Natural Language Query Interface to Structured Information.