

Markov Logic for Ontology based Information Extraction

M. D. S. Seneviratne

Submitted in partial fulfillment of the requirements for the degree of
Doctor of Philosophy
University of Colombo School of Computing
2019

Declaration

The Thesis is my original work and has not been submitted previously for a degree at this or any other university/institute. To the best of my knowledge it does not contain any material published or written by another person, except as acknowledge in the text.

Author's Name Mrs M. D. S. Seneviratne

Date


Signature 

This is to certify that this thesis is based on the work of Mrs M. D. S. Seneviratne under my/our supervision. The thesis has been prepared according to the format stipulated and is of acceptable standard.

Certified by

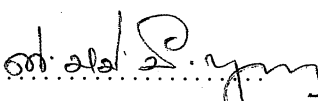
Supervisor 1 Name Dr D. D. Karunaratne

Date

Signature 

Supervisor 2 Name Dr (Mrs) K. S. D. Fernando

Date

Signature 

Acknowledgments

First I would like to specially thank Dr. D.D. Karunaratne and Dr(Mrs). K.S.D. Fernando for agreeing to supervise my research at the transition of MPhil. to PhD. Completion of this research would not have been possible without their support and feedback. Further, I thank Dr. D. D. Karunaratne for the guidance given at the conversion and throughout the research. I am especially grateful to Dr. K.S.D. Fernando for taking time in regular discussions to monitor the progress of my research and guiding me in the right direction. I am thankful to my first supervisor Dr. D. N. Ranasinghe for his tremendous support, advice and guidance given at the initiation and continuation of the research.

I am much obliged to Prof. A. Karunananda for discussing the initial research proposal, giving advice and research ideas on various occasions. I also thank Prof. A. Ginige for taking time out of his busy schedule to discuss the aspects of my research during his short stay in Sri Lanka and assisting me with resources for a test bed.

I sincerely remember Late Senior Professor Gihan Wikramanayake for making necessary arrangements to proceed my request for MPhil to PhD conversion.

I cannot wind up without thanking my husband Gihan Seneviratne for persuading me to proceed with the research. Without his encouragement I would not have pursued this research.

Abstract

Today's world, Internet has become a fast and efficient information provider although the relevancy or accuracy of the information found is not guaranteed. Web itself presents numerous problems mainly due to its heterogeneous nature with respect to semantics and representation formalisms, high dynamicity and the overwhelming size of the resources available. Therefore, encoding semantics of web documents in a formal way is a necessity in effective information gathering. Ontology plays a vital role in enhancing the semantics of natural language documents in machine-readable form on the semantic web. In ontology construction and in linking terms in web documents with appropriate concepts in Ontologies, terms and relationships between them should be extracted. Successful information extraction, for ontology construction needs to focus on natural language sentences for identifying the concepts the document entities represented and their relationships. As a result of the efforts made by semantic web researchers, numerous established techniques and tools are available mainly for the extraction of entities which form basic constituents of ontology, the concepts. However, the associated Relation extraction is yet to be addressed extensively. Therefore, the present work concentrates on extracting domain specific entities and generating relation-extraction-rules to extract relations for ontological structures. The presented method exploits the existing techniques for entity extraction and introduces a novel approach for relation extraction based on a set of rules specifying the dependencies of entities in natural language sentences. Adaptation of Inductive Logic Programming to generate relation-extraction-rules from language dependency clauses and modeling them on Markov Logic Network environment for statistical relation extraction by using a domain independent approach, distinguishes the present work from previous work in the area of information extraction.

The evaluation of the system shows the effectiveness of the proposed method in domain specific information extraction and in document classification as a proof of concept. Document classification shows a high accuracy in all classifications especially with 100% precision on number of occasions in the selected domain. Furthermore, this research stresses the importance of evolving training corpus automatically in order to minimize the manual involvement in supervised learning in creating a large amount of training data.

List of Acronyms

OWL	: Web Ontology Language
MLN	: Markov Logic Network
CD	: Contrastive Divergence
MCMC	: Monte Carlo Markov Chain
ILP	: Inductive Logic Programming
LINUS	: Attribute value learning system
POS	: Part Of Speech
DHDB	: Deductive Hierarchical Database
NN	: Noun
VB	: Verb
KNN	: K-Nearest Neighbour
SVM	: Support Vector Machine
MAP	: Maximum a Posteriori
CI	: Class Index
RI	: Relation Index
MBE	: Microaveraged Recall/Precision Break Even Point
ABE	: Macroaveraged Recall/Precision Break Even Point

Contents

Chapter 1 Introduction.....	1
1.1 Background	1
1.2 Motivation	3
1.3 Aims and Objectives.....	7
1.4 Approach towards Solution and Contributions	8
1.5 Outline of the Remaining Chapters	10
Chapter 2 Background Theory	13
2.1 Introduction	13
2.2 Semantic Web and Ontology	14
2.2.1 Web as a large data repository.....	14
2.2.2 Enabling the Semantic Web	15
2.2.3 Application of Ontology on the Semantic Web	16
2.3 Information Extraction from the web for Ontology	
Construction/Population.....	17
2.3.1 Role of Entities in Ontology.....	17
2.3.2 Relations in Ontology.....	18
2.3.3 Common methodologies used in entity and relation extraction	20
2.4 Natural Language Processing	20
2.4.1 Complexity of Natural Language Text	21
2.4.2 Natural Language Parsing and Dependencies	21
2.5 Rule based Techniques	22
2.5.1 Rule Learning Algorithm	23
2.5.2 Markov Logic Network for Statistical Relation Extraction	24
2.5.3 GATE's Information Extraction System ANNIE.....	27
2.6 Ontological Information Extraction for Document Classification	28
Chapter 3 Related Work	30
3.1 Introduction	30
3.2 Using Rules in Information Extraction	30
3.3 Natural Language Processing based Techniques	34
3.4 Statistical Methods for Ontological Information Extraction	38
3.5 Related work on Text Classification	41
3.6 Weaknesses and Problems Identified from the Literature Review.....	46

Chapter 4 Ontological Information Extraction.....	48
4.1 Introduction.....	48
4.2. Ontological Entity Extraction	49
4.2.1 Extending GATE's facilities with additional components.....	49
4.3 Natural Language Parsing Towards Relation Exaction	54
4.3.1 Reducing Stanford Dependencies for for Relation Extraction.....	55
4.3.2 Identifying Conjunctions and Relative Clauses	62
4.4 Inductive Logic Programming for Generation of Relation-extraction-rules..	63
4.4.1 Taxonomic Relations.....	65
4.4.2 Non Taxonomic Relations.....	65
4.4.3 Extraction of Ontological Relations.....	67
4.4.4 Adaptation of Attribute Value learning for Ontological Relation.....	68
4.4.5 Processing the training set.....	71
4.4.6 Processing positive and negative training data using the attribute value Learner.....	72
4.4.7 Weakening the Language Bias.....	78
4.5 Markov Logic Network for Statistical Relation Extraction.....	79
4.5.1 Applicability of MLN on Relation-Extraction-Rules.....	80
4.5.2 Sampling Atoms for MLN.....	82
4.5.3 Weight Learning for Relation-Extraction-Rules.....	84
4.5.4 Weight Optimization.....	86
 Chapter 5 Use of Relation-Extraction-Rules on Document Classification.....	 89
5.1 Introduction.....	89
5.2 Document Classification.....	89
5.2.1 Representing Documents in Entity-Relation framework.....	90
5.2.2 Determination of Classification Performance.....	90
5.3 Comparison of the Proposed Document Classification Method with other Related Work.....	92
5.4 Expanding the Training Corpus.....	97
 Chapter 6 Implementation.....	 102
6.1 Introduction.....	102
6.2 Extending GATE for domain entity extraction.....	102
6.2.1 Entity Identification for the selected domains.....	104
6.2.2 Identification of Patterns for Selected Entities.....	107
6.3 Design of the Generation of Relation-Extraction-Rules.....	109
6.3.1 Generation of Relation-Extraction-Rules.....	110

6.3.2 Ontological Relation Extraction.....	116
6.4 Document Classification.....	118
 Chapter 7 Experimental Results and Performance Evaluation.....	119
7.1 Introduction.....	119
7.2 Data Sets and Performance Matrices.....	120
7.2.1 Data Sets for the Experiments.....	120
7.2.2 General Evaluation Criteria and Quality Measures.....	122
7.3 Evaluation of Entity Extraction.....	124
7.4 Relation Extraction.....	126
7.4.1 Extracted Relation Instances and New Relations.....	126
7.4.2 Evaluation of Relation Extraction.....	129
7.5 Evaluation of Document Classification by Relation-Extraction-Rules.....	133
7.5.1 Evaluation based on the Selected Domain.....	133
7.5.2 Evaluation based on a Benchmark-corpus widely used for Text Classification.....	135
 Chapter 8 Conclusions and the future work.....	141
8.1 Summery.....	141
8.2 Conclusions.....	142
8.2.1 Processing Natural Language Sentences towards Identification of Relations.....	142
8.2.2 Generation of Rules for Extraction of Relations Embedded in Natural Language Sentences.....	143
8.2.3 Assigning Weights for Relation-Extraction-Rules.....	144
8.2.4 Testing the Applicability of Relation-Extraction-Rules on a Classification Task.....	145
8.2.5 Using Dynamic Training Corpus on Supervised Learning.....	147
8.2.6 Overall Concluding Remarks	148
8.3 Future Work	149
 Bibliography.....	160
 Appendices.....	160
Appendix A - Fundamentals of Ontology.....	A1
Appendix B - Summary of the related work.....	A3
Appendix C - ANNIE- GATE's Information Extraction System.....	A11

Appendix D - JAPE rules and creole for the entity extraction in domain “Bird”.....	A15
Appendix E - Part-of-Speech Tags used in the Hepple Tagger.....	A22
Appendix F - Definitions of Stanford Dependencies.....	A24
Appendix G - Samples of Reduced Dependencies for the Relations in the Domains Bird and Sport.....	A29
Appendix H - Relation Extraction Rules.....	A43
Appendix I - Relations and Relation Instances found by the Rule-based Systems.....	A55
Appendix J - Semantic Patterns for Entities in the Corpus Reuters-21578.....	A70

List of Tables

4.1 Commonly used Conjunctions.....	62
4.2 Examples of Combinations of Atomic Formulas for the Relation <i>located_in</i>	73
4.3 Some Positive and Negative instances for the Relation <i>located_in</i>	74
4.4 Generalization of the Data shown in Table 4.3.....	75
4.5 Possible Groundings and Evidence.....	81
4.6 State Space of the Markov Blanket of an atom.....	85
5.1 Examples of Relations and respective Entity tuples.....	92
5.2 Summarization of the comparison of proposed approach with other related methods.....	96
5.3 The range of RI for each sentence category.....	101
6.1 Token types and possible attributes and values.....	103
6.2 Main Entities in the domain Bird	105
6.3 Main Entities in the domain Sport	107
6.4 Identified Patterns for some Entities in the domain Bird	108
6.5 Identified Patterns for some Entities in the domain Sport.....	109
7.1 Evaluation Measures for the basic Entities of the domain Bird	122
7.2 Evaluation Measures for the basic Entities of the domain Sport.....	124
7.3 Comparison of three other systems with presented system.....	125
7.4(a) Results of the relation type <i>located_in</i> ().....	127
7.4(b) Results of the relation type <i>related</i> ().....	128
7.4(c) Results of the relation type <i>played</i> ().....	128
7.5 Statistics of Extraction of instances for the relation <i>located_in</i> ().....	129
7.6 Evaluation Measures for Relations from two domains	132
7.7 Comparison of performance of Relation-extraction-rules	132
7.8 Comparison of the Precision with two different approaches for two relations....	133
7.9 Classification Performance with respect to N and CI in the domain Bird	134
7.10 Sub classification of the main class Bird on Bird type and Eat type.....	134
7.11(a)Category <i>acq</i>	136
7.11(b)Category <i>dlr</i>	136
7.11(c)Category <i>bop</i>	136
7.11(d)Category <i>trade</i>	137
7.11(e)Category <i>earn</i>	137
7.11(f)Category <i>ships</i>	137
7.11(g)Category <i>jobs</i>	138
7.12 Classification performance of the selected categories of Reuters-21578	138
7.13 Comparison of the proposed method	138

List of Figures

1.1 Overview of the System	12
2.1 Fraction of Ontology for two Entities <i>Bird</i> and <i>Location</i>	18
4.1 GATE's User Interface with an annotated document	52
4.2 Part of a Text from GATE's Output annotated with the Entity <i>Location</i>	53
4.3 GATE Output saved as a XML file.....	53
4.4 Graphical Representation of Stanford Dependencies for a Sentence	56
4.5 Graphical Representation of Reduced Dependencies of a Sentence	60
4.6 The Process of generating Reduced Dependencies from Stanford Dependencies of an English Sentence	61
4.7 Main Sentence Categories with respect to Information Extraction	63
4.8 (A) Initial Rule, (B) Modified Rule, (C) The Final set of Rule	69
4.9 Input and Output of the two processes Rule Learning and Relation Extraction...	70
4.10 The Network of the grounded atoms with respect to Rule 1.....	82
4.11 Reduced Network after removal of Evidence	83
4.12 The Network when the Verbs are replaced with one equivalent verb	83
4.13 Markov Blanket of an atom	85
4.14 Overview of the Weight Learning Process	88
6.1 Abstract View of the Rule Learning Process	115
6.2 Abstract View of the Relation Extraction Process	117
7.1(a) Probability of Relation Extraction in the domain <i>Bird</i>	130
7.1(b) Probability of Relation Extraction in the domain <i>Sport</i>	130
7.2 Recall/Precision performance of fully and partially annotated documents	135
7.3 Recall/Precision performance of 7 categories in the Reuters-2578	139

Chapter 1

Introduction

1.1 Background

At present, the web has become a major information source for many people. The web being the latest and fastest information provider, people prefer accessing it for their information needs. However, finding a specific piece of information from a massive collection of web sources is a tedious, time consuming task for a human being. Therefore semantic web researchers have made numerous efforts in making web pages machine readable by annotating the text in web pages with semantic tags and developing ontology to model the information in a more structured manner [2, 27, 33]. Ontology development has emerged as a means of standard representation of various types of web pages in the same domain. Ontology contains concepts in a domain of discourse and relationships among them. A concept represents a class of entities and relations can represent also properties of concepts describing various features and attributes of them. Various tools [93, 100] available at present for ontology construction require basic building blocks which are the domain specific entities and their relationships in order to create ontology. Information extraction, concept definition from various web sources and text mining are required processes for identifying entities and relationships. Therefore these processes have been widely investigated for ontology development.

A considerable amount of work has been carried out in the area of information extraction at a preliminary stage. Many researchers have exploited machine learning [1, 8, 11, 12, 13, 20, 32], pattern matching [7, 18, 48], shallow natural language processing [3, 5, 8] and statistical methods [11, 21, 35]. However, further improvements are required to increase the precision for successful information extraction. Machine learning is the main technique adopted in information extraction process. Statistical machine learning methods such as Support Vector Machines [96, 98] Hidden Markov Model [99] etc. as well as rule based learning [1, 5, 8, 12, 13, 14, 30, 45, 50, 51] have also been exploited extensively in research work.

Supervised, unsupervised, semi supervised methods and distant supervised methods are used in information extraction. Supervised methods achieve higher accuracy at the expense of using a vast amount of labeled training data. Performance measures are reportedly low in unsupervised methods though it avoids the cost of using large amount of training data. To overcome the disadvantages of supervised and unsupervised methods, research work begins to focus on semi supervised and distant supervised methods. Distant supervising is based on large knowledge bases such as the freebase available on the web. Although the use of a large public knowledge base may be effective on general information gathering its efficiency in extraction of domain specific information cannot be guaranteed. Therefore semi supervised methods which use rather small set of labeled training data at the beginning have the advantages of both supervised and unsupervised methods, overcoming their disadvantages to a certain extent.

Furthermore, work in identifying relations between concepts which is more complicated has not yet been progressed satisfactorily. Relation extraction requires heavy linguistic processing of a given text and needs to be addressed in order to complete the information extraction process. Therefore despite the effort made by the researchers, finding ontological information from unstructured text still remains a complicated task which requires new and refined techniques. Mining the web for finding relevant information sources is the first step to be carried out in information extraction. Assigning text documents to predefined classes of documents online, is considered as an effective approach to finding useful information from numerous online text repositories. Therefore document classification can play an important role in fetching the domain specific documents from the web. It finds a variety of applications including information extraction, news filtering & organization, document organization & retrieval, opinion mining & sentiment analysis and e-mail classification & spams filtering [81]. However, the exponential growth of web resources and the higher dimensionality of documents make the automation of document classification a huge challenge for the data mining community. Once the documents are categorized into their respective classes representing different domains, methodologies can be used on each domain to extract specific information. In this regard a number of document classification methods can be found; Naive Base classification [64, 81], support vector classification [67, 68], decision trees

[81] and rule based classification [79, 82] are such popular techniques. However, since most of these methods use word counting and word vs. document proportion as features, these features used to represent documents should be selected before applying those classification techniques. In general, a document is mainly represented by the concept of bag of words where a set of words together with their frequencies are used to represent the document or as a string where the document is represented by a sequence of words. In addition techniques such as rough set, principle component analysis etc. [81] are applied in order to find the minimum set of features without significant loss of information. All of these techniques require a considerable amount of effort to find the most relevant features from a document to be used in the selected text classification method and mostly end up with a large number of terms with noisy irrelevant features. Therefore the above mentioned methods being widely used popular techniques in text classification, information extraction can be considered as a potential alternative area to be researched for document classification.

1.2 Motivation

As explained in above section 1.1 ontology is a strong representation that bridges the gap between the semantic web and the unstructured natural text. The machine can access the ontology and provide information required by users, saving the user from the laborious task of searching numerous web sources and surfing through the jumble of natural text to find a piece of information. Reliability of ontology depends upon the accuracy and timeline of information provided by it. Ontology development requires identifying entity classes, class instances, taxonomic relations to accommodate sub classes and non-taxonomic relations to define properties of the classes and establish relationships between entities. Therefore correct identification of above mentioned information is a crucial factor in successful ontology development. However, extracting information for ontology development from various natural language sources is a time consuming tedious task. Since the natural language text is vast in terminology and sentence patterns, extracting information wrapped in natural language sentences in order to model them into the structured format of ontology is a complex and continuous process. Therefore many researchers [3, 5, 7, 15, 16, 27, 35, 38, 40, 43, 51] have focused on automating/semi

automating the ontology development process. They have concentrated in their efforts on extracting entities, entity instances and taxonomic relations in creating/populating ontology, but relation extraction still remains more challenging. In addition, many systems are based on domain ontology and adapting the system to work in different domains demands heavy manual involvement. Therefore the extraction of non-taxonomic relations has been identified as the main problem to address in the research work presented in this thesis on ontology based information extraction.

Although the basic entity extraction has been addressed widely, almost all such systems face the problem of extraction of irrelevant entities (i.e. false positives) and few have suggested and implemented some techniques [7, 10, 21, 30, 45, 51, 53, 57] to filter out irrelevant information. Most of them [7, 10, 21, 30, 45] have used dictionaries, semantic gazetteers and other web services [10] to confirm the extracted entities. Yet it has not proved to be very successful and success highly depends on the application domain. False positives will populate ontology with incorrect entity information which is more harmful than lack of information. As such with the contributions researchers have made towards information extraction, it is still necessary to find methods in order to minimize extracting irrelevant information.

Since Rule based systems are declarative and easy to comprehend, maintain and incorporate domain knowledge [84] these systems are widely used in information extraction as mentioned in section 1.1. Under a supervised rule based approach a set of rules is generated from training data. Some system use hand coded rules [51] and some systems use machine learning algorithms to induce rules from training data [1, 5, 12, 13, 30]. The antecedent of the rules used to extract relations, contains the condition which should be satisfied for the relation to be true. The rules need to be weighed to reflect their strength which contributes to finding the probability of an extracted relation instance. Therefore finding the weight of a rule is very necessary in determining the accuracy of the extracted information. However, most of the previous rule based information extraction systems either lack a weight learning process [1, 8, 12, 13, 50] or employ a poor weight learning method [5, 36, 51]. This implies the necessity of a proper weight learning process in rule based systems. Markov Logic that accomplishes weight learning

for first-order formulas can be investigated for the possibility of weight learning in rule based information extraction systems and hence for statistical relation extraction.

Although statistical machine learning has become the choice of many recent academic researchers in information extraction, rule based methods find a higher applicability in the practical environment and dominate the commercial world [84]. Therefore hybridizing rule based systems with statistical machine learning is a promising initiative for improved information extraction systems. Inductive logic programming and Markov Logic Network can provide a good foundation for a statistical machine learning approach in a rule based system for relation extraction.

In rule based classification the antecedent of the rule contains the condition which relies on the feature set while consequence defines the most possible class label. Normally the condition consists of a pattern of word combinations and terms. Therefore a large number of rules can be generated for a predefined class but the rule based methods suffer from irrelevant noisy features and large number of rules. Two of the most commonly used criteria in rule generation are those of support and confidence [81]. Support indicates the number of instances in the training set which are relevant to the rule and Confidence is the conditional probability that an instance in the training set belongs to a class given by the rule when the condition is satisfied. However Support does not give a clear indication of the strength of the rule whereas Confidence is a more direct basic measure of the rule strength. Thus, the Confidence is a better criterion only in comparison to Support. Yet both Support and Confidence are widely used measures in ordering and refining the rule set. When a test instance satisfies a number of rules with the same class label a class can easily be assigned to the test instance, but when the satisfied rules are relevant to different classes the above mentioned confidence measure is used for conflict resolution. Since the measures Support and Confidence do not normalize for a prior presence of different terms and features, the classification rules are prone to misinterpretation on training data corpus with imbalanced class distribution. When a document class is signified by a large number of features and rules, confidence based conflict resolution might not be sufficient for accurate classifications. This emphasizes the requirement of more sophisticated techniques for the selection of class specific features and for conflict resolution. When considering the problems associated with text classification this work concentrates on

investigating the possibility of using the same techniques developed for information extraction in document classification as a proof of concept. Thus, both information extraction and document classification can be addressed simultaneously in the same system which makes document classification an application of information extraction.

By paying attention to drawbacks in purely supervised and unsupervised learning methods, the presented work is focused on using a rather small set of labeled training text corpus in the semi supervised manner and expanding the corpus automatically by the system itself. Then the information extraction process can be repeated on the expanded training corpus for further improvement until it comes to a static state with respect to the improvement on the system.

After considering above mentioned points the following research questions are identified and investigated.

- Processing natural language sentences for identification of relations
 - Which language characteristics are most effective in rule based relation extraction?
 - How can unwanted information in a sentence be removed to focus on extracting the relation embedded in the sentence?
 - How can lengthy sentences with conjunctions be processed to identify potential relationships?
 - Do the selected language characteristics need refining to be used in ontological relation extraction?
- Generation of relation-extraction-rules for extraction of relations embedded in natural language sentences.
 - What is the type of training data used in generating relation-extraction-rules with selected language characteristics ?
 - How can relations be defined and the language characteristics be used in formulating relation-exaction-rules ?
 - How can machine learning be used in inducing relation-extraction-rules from training data to achieve high performance measures by avoiding the extraction of false relation instances ?
- Assigning weights for the relation-extraction-rules for statistical relation extraction
 - How can the relation-extraction-rules be modeled in the statistical environment for

weight learning ?

- How can the weights be properly assigned to measure the strength of the rules ?
- Testing the applicability of relation-extraction-rules on a classification task as proof of concept.
 - Is it possible to address the problems encountered with the state of art classification methods by using ontological information ?
 - How can the relations-extraction-rules be used in document classification for improved performance?
- Using dynamic training corpus on supervised learning.
 - How can the extracted information be used to expand the training corpus automatically to avoid the time and manual labour consumed in creating a large training corpus ?

Since there are techniques and tools already established for language processing tasks, an existing tool can be used for entity extraction with possible improvement. Therefore entity extraction is not addressed as a potential research question here.

1.3 Aims and Objectives

The aim of the present research is to develop techniques for extracting information for ontological structures addressing the above mentioned research questions.

The main objective of the research is to acquire knowledge for ontology construction from a massive collection of information sources on the web by analyzing and processing the web documents. This implies

- investigating the use of existing tools for ontological information extraction. This investigation will lead to finding out the applicability of existing tools in discovering the relevant domain concepts and adapting the selected tool for the purpose by exploiting its advantages and minimizing disadvantages.
- developing a rule based methodology to use for relation extraction incorporating statistical machine learning
- designing a weight learning system for relation-extraction rules developed.

- applying the developed information extraction system on several domains and evaluating the obtained results in order to analyze the suitability and performance of the system.
- investigating the possibility of applying the methodology developed for relation extraction in document classification with a critical comparison to other well established text classification methods.
- developing a technique to use the relation-extraction rules generated in document classification
- employing the methodology developed for information extraction in document classification
- evaluating the performance of the system on document classification
- expanding the training corpus with suitable extracted information

1.4 Approaches towards Solution and Main Results

The initial work of the research is concentrated on identifying entities in a web based document followed by extracting relations for the identified entities. As it was emphasized on the requirement of addressing relation extraction extensively in previous sections a higher weight is given for it in the research. Therefore, this work first focuses on extracting ontological entities from domain specific text documents. Then it fully concentrates on extracting relations existing between extracted entities by using a rule based system which incorporates statistical machine learning.

Since the focus of the research is not entity extraction, effort is made to use existing techniques and tools for entity extraction. Rules written in JAPE are packaged into GATE (General Architecture for Text Processing) for domain specific entity extraction.

Once the documents are annotated with entities and relations, annotated sentences are parsed using Stanford parser to obtain dependencies of each sentence. Dependencies of sentences are processed in order to filter out unnecessary dependencies which do not make any contributions to the relations present in the sentence. Then the dependencies of a sentence are reduced and relation-extraction-rules are generated from the reduced dependencies by using Inductive Logic Programming technique. Evaluating of domain specific relation extraction is done assuming 100% accuracy in entity extraction phase

because entity extraction and relation extraction are done separately by two different processes. Any annotated sentence from which a relation instance cannot be identified for an existing relation is assumed to be a candidate for a new relation.

The relation-extraction-rules are modeled in Markov Logic Network (MLN) to determine the weight for each rule. Weight learning requires the use of optimization techniques. In weight learning process the set of initial rules with a maximum of three clauses are considered due to intractability of having many clauses in the rules. It is assumed that there won't be a significant weight difference between the initial rule and the final rule which contain more clauses to avoid the extraction of false relation instances. Justification for the assumption is given in chapter 4 section 4.5.1.

The extracted ontological entitles and relations are used in document classification in order to test the applicability of extracted entities and weighted relation-extraction-rules in document classification. Figure 1 shows the overview of the entire process.

Contributions of the research are given below.

- Processing resource components for GATE [14] to extract entities in the test domains.
- Reduction method to retain only the necessary information relevant to relations present in a sentence, from the language dependencies of the sentence.
- A statistical machine learning method to generate rules for ontology based relation extraction with Inductive Logic Programming for rule induction and Markov Logic Network for weight assignment for the rules and probability assignment for extracted relation based on rule weights. The resulted relation-extraction-rules are capable of extracting instances for new relations as well as for known or predefined relations existing between known entities in a selected domain.
- Evaluation of entity extraction and the performance of relation-extraction-rules on the selected test domains with a comparative analysis of the results with relevant previous work on information extraction.
- Documents classification method based on relation-extraction-rules.
- Evaluation of the document classification method on test domains used in the information extraction and on a bench mark corpus used in text classification

followed by a comparison of the performance on the bench mark corpus with previous work.

- A method to automatically expand the training text corpus.

1.5 Outline of the Remaining Chapters

The rest of the theses are organized to explain background theory and a literature review followed by our contribution towards ontological information extraction and document classification with concluding remarks.

Chapter 2 mainly explains the background theory related to semantic web, ontological information extraction and document classification. First, it explains the problems encountered with the World Wide Web as a large information repository and how the concept of semantic web can be an answer to those problems. Discussion is continued on ontology as a formal representation for information sources and extraction of required information for ontology construction. Theoretical aspects which are used in developing ontological relation extraction system and the tool used for entity extraction are described here. The importance of document classification in information extraction and the possibility of using information extraction in document classification are also discussed.

Chapter 3 identifies rule based techniques, natural language processing techniques and statistical methods as main methodologies used in information extraction from web information sources and gives a literature survey under each methodology. Furthermore, critical discussion on widely used techniques in text classification and related work in the area by highlighting the main draw backs with the already existing techniques is given in this chapter. Finally weaknesses and problems identified in the literature review are presented.

Chapter 4 describes the techniques developed for ontological information extraction based on first three research problems listed in the chapter 1 section 1.2. The present approach in generating additional components for GATE in order to extract domain specific entities is explained first. Then the relation extraction process is described. The

possibility of reducing the dependencies produced by Stanford parser for generation of relation extraction rules is described here. Next, the process of using inductive logic programming to generate relation-extraction-rules from reduced Stanford dependencies of sentences, annotated with domain specific entities, is described with examples. The chapter concludes with a detailed description of the use of Markov Logic Network (MLN) on learning weights for relation-extraction-rules.

In the chapter 5 the methodology for document classification is described based on relation-extraction-rules. This approach is discussed and reviewed with respect to already existing well established text classification methods. Finally measures to use for training corpus expansion are also discussed here

Chapter 6 gives a detailed description of the implementation of both information extraction and document classification methodologies. Further, the results of document classification on the bench mark corpus Reuters-21578 is also shown in this chapter.

In the Chapter 7 the definition of the measures used to quantify the quality of the results is presented followed by an evaluation of the results. Comparison of the results in entity extraction with three other systems; Armadillo [10], Amilcare [8] and Ontoshopie [5] are also shown in this Chapter. For relation extraction a comparison made with T. Wang's methodology which also involves GATE is presented. Comparison of two individual relation types with two other systems is also given in this chapter. This chapter shows the performance of the proposed document classification method on two types of test data and concludes with a comparison of the method with a previous research on the same bench mark corpus Reuters-21578.

Finally, the Chapter 8 contains concluding remarks of the present work and directions for possible future work.

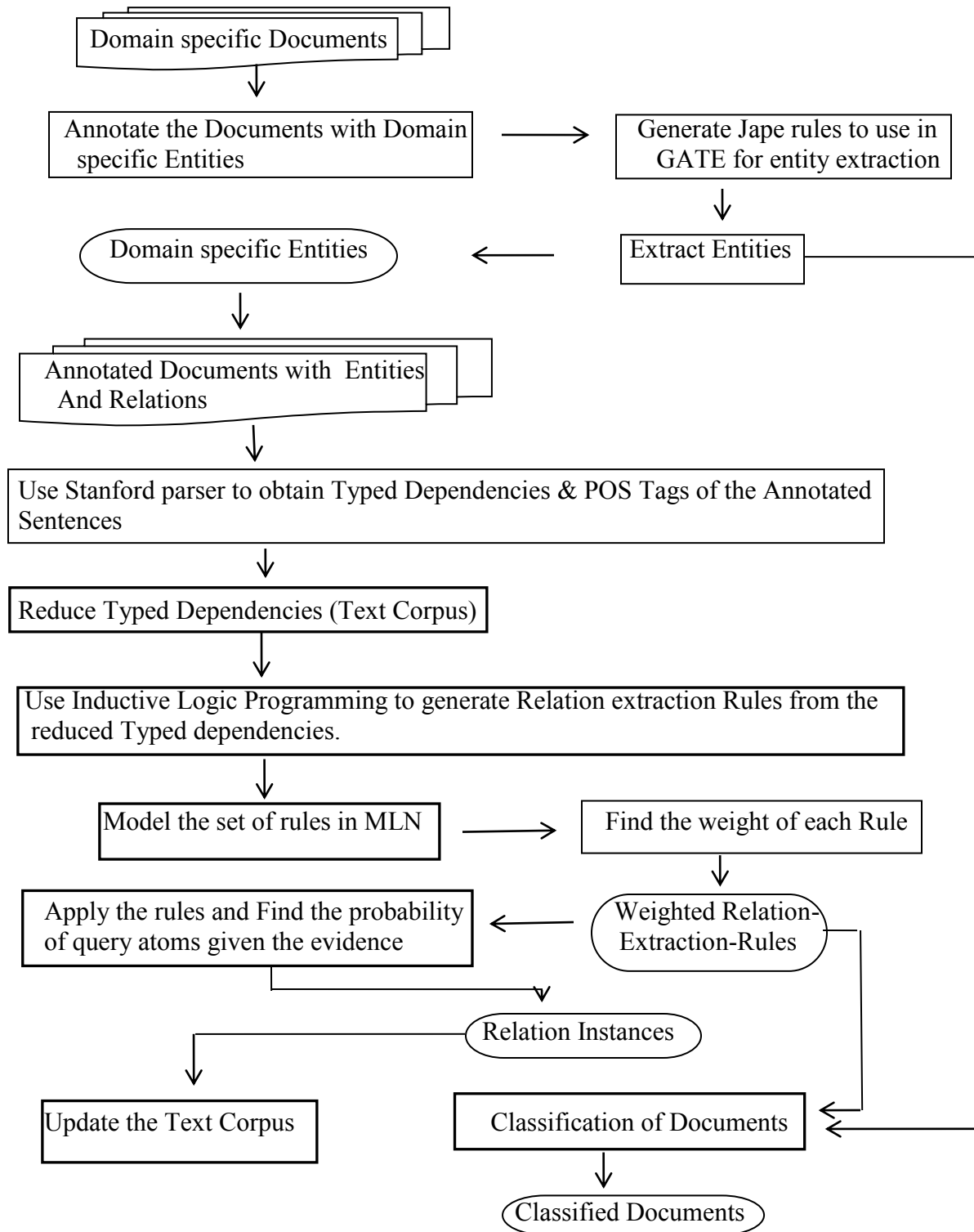


Figure 1.1 Overview of the System

Chapter 2

Background Theory

2.1 Introduction

Ontological information extraction which is the focus of the research work presented in this thesis is inspired by already available methodologies & tools in the area of semantic web and information extraction. This chapter gives a description of theoretical aspects and tools relevant to the research. It includes a study of web characteristics with respect to information extraction and the main concepts already established towards web information extraction. The chapter concludes discussing the applicability of ontological information extraction on document classification

Section 2.2 describes the World Wide Web as a rich source of information and problems associated with retrieving required information from it. Then it explains the concept of semantic web and how it can be tailored for user requirements. It also mentions the difficulties associated with the semantic web and how can the concept of ontology be used to overcome them. Fundamentals of ontology are given in Appendix A. It further explains information location and extraction as the major challenge in automating ontology construction and population.

Section 2.3 begins with an introduction to entities and a description of how the entities occur in ontology. Entities alone are not meaningful if they are no means to connect them. Therefore this section explains the existence of relations between entities in ontology. This section also states common methodologies employed for entity and relation extraction over the past years.

Natural language techniques applicable to current research work presented in this thesis is described in the section 2.4.

Section 2.5 explains rule based techniques along with a statistical method used to model rules for statistical relation extraction. This section also includes a description of an information extraction tool in already established text processing tool GATE.

Section 2.6 gives a brief description of the importance of combining information extraction with document classification.

2.2 Semantic Web and Ontology

The Web requires a human to assimilate and manipulate the multiple pieces of information available. Such information based tasks involve users in different information views and a set of actions such as filtering, saving, searching, extracting, classifying and merging etc. to be performed on the found information. A human may find that a useful piece of information is spread out over multiple sources, making the seeking of information task a very strenuous process. Therefore the need for a machine understandable and queriable information and knowledge layer has become vital [4]. Semantic web which can be considered as an extension of the conventional Web (Web2.0), attaches meanings to the information rather than solely displaying them. Then the Semantic Web should be able to answer user's queries and provide meaningful information by deducing facts from a large pool of unstructured information. Therefore the challenge of the Semantic Web is to provide a language that is expressive enough for both data and rules to reason about data which allows the rules from any existing knowledge representation, to be brought in to the Web. In order to enable the concept of semantic web the problems associated with the conventional web should be addressed.

2.2.1 Web as a large data repository

At present the Web has become a major information source for almost every possible domain of knowledge. Therefore the web can be considered as a valid repository for information location and knowledge acquisition and being the latest and fastest information provider, people prefer accessing the Web for their information needs. Information searching from the Web has one or more dimensions that may include a technical component (e.g. search algorithm), an organizational component (how information is structured) and psychological component (how information is presented so that humans can easily find what they are looking for) [33]. But there are significant problems associated with the World Wide Web with regards to above mentioned components. The research work described here focus on the issues related to organizational and psychological components. Therefore the followings can be outlined as a series of problems raised by the web that should be tackled in finding accurate reliable information from this source.

- (a) Web resources are presented in natural language and contain a vast amount of information. This makes the web resource noisy and valuable information is often overlooked. User also might not know how to specify the information that he wants. Therefore unstructured nature of the natural language adds stress to the user in searching for a piece of information that he requires.
- (b) The enormous amount of information available on the web overwhelms the users. Hence finding a specific piece of information from the massive collection of web sources is a tedious, time consuming task for a human being.
- (c) Web sources are being continuously updated in a dynamic way and make any attempts difficult in structuring the information available on the web. A human being might find it impossible to cope with these rapid changes.
- (d) The accuracy of the information available on the web is not guaranteed and some information may be contradictory leaving the user in confusion.

The proposed work here takes attempts to address the first three issues (a), (b) and (c) by means of making web pages machine readable. Since extracting ontological information from individual web resources and is focused assuming that the available information on the Web is accurate, attempting to make rectifications to available information on the Web is beyond the scope of this research. Therefore the problem mentioned in above (d) is not addressed in this project.

2.2.2 Enabling the Semantic web

Extensible Markup language (XML) and Resource Development Framework (RDF) [95] are two technologies already in place for developing Semantic Web concepts. XML allows users to annotate their document with arbitrary tags but does not provide facilities to indicate the underlying meaning of the tags. Meaning of information can be expressed in RDF by encoding it in sets of triplets which are similar to subject, object and verb of an elementary sentence [95]. These triplets can be written using XML tags and subject and object are each identified by a Uniform Resources Identifier (URI) located somewhere on the web. The high flexibility of natural language makes room for the same term to be used for somewhat different meanings and two or more terms to be used for the same meaning but a formal representation cannot accommodate such flexibility.

Therefore Semantic web researchers have made numerous efforts to make web pages machine readable by creating wrappers for web based information sources, annotating the text in web pages with semantic tags and developing ontologies to model the information in a more structured manner[12, 13, 42]. Wrappers contain a set of extraction rules suitable to extract information from a web site.

2.2.3 Applications of Ontology Concept on the Semantic Web

The concept of Ontology [Appendix A] is a better alternative for formal representation of information extracted from various information sources in order to address the above mentioned issues as ontology population is a continuous process and the users can conveniently search through the ontology space to obtain up to date information. Therefore many researchers have focused on constructing ontologies to address the problems in searching information from the enormous web resources [4]. Ontologies can also be used to annotate web pages with semantic tags. Sometimes tags are created in order to build the ontology.

Ontology finds wide applications on the Web although it is not limited to the Web. The heterogeneous nature of domain resources requires a sharing of common understanding of the structure of information. Ontology appeared as a response to this requirement. In addition, ontology can be considered as a standard model of domain knowledge that separates domain knowledge from operational generic knowledge [6]. Many disciplines have now developed standardized ontologies that domain experts can use to share and annotate information in their fields [19, 46, 47, 48, 90, 91, 92].

Ontology can provide the solution for the problem mentioned in the above section 2.2.1 by defining relations among terms. Entities and relations play a powerful role in information extraction. Some relations among entities can be expressed by assigning properties to classes and allowing sub classes to inherit such properties. Inference rules in ontologies can readily deduce facts from available entities and relations. The use of equivalence relation in ontology provides an answer to the problem of the use of more than one term for the same meaning. This further strengthens the manipulation of web resources much more effectively in ways that are useful and meaningful to the human user.

2.3 Information Extraction from the web for Ontology

Construction/Population

Construction of ontology involves finding concepts and their relations from massive heterogeneous information sources [43]. Manual construction of domain ontologies is therefore a laborious and expensive task. Finding all the possible domain specific concepts and relations is strenuous and time consuming. Since the information source is bound to get expanded over time it is difficult to maintain an up to date ontology. For e.g. Gene Ontology, a prominent ontology in biology consists of more than 28000 terms describing molecular function, biological processes and cellular locations of genes [87]. In the last few years a number of tools for ontology construction have been introduced to the ontology arena [35, 89, 93, 100]. But those tools require to be provided with data for the ontology construction and provide means for manipulation of data in order to import to an ontological framework. None of the tools is powerful enough to gather and process ontological information from various sources effectively. Therefore extracting domain specific data has become the biggest challenge in constructing or populating ontologies. Since the basic elements of ontology are entities and relations, extracting them from web resources is the initiative step towards the automation of ontology construction.

2.3.1 Role of Entities in Ontology.

Entities are embedded in noun phrases and can comprise of one or more tokens from the unstructured set. Entities can normally be of various types and the entities of same type can be divided into sub classes. For e.g. the entity *person* can be an *employee*, *student*, *patient*, *customer*, *sportsman*, *actor* etc. The most general and popular type of entities is Named Entities such as person's names, locations, organizations etc. Named Entity Recognition was first introduced in the 6th MUC [101] and consists of three categories: proper names and acronyms of persons, locations, organizations (ENAMEX), absolute temporal terms (TIMEX) and monetary and other numeric expressions (NUMEX). Also any noun specific to a domain is captured as an entity. For e.g. bird's name, diet, habitat etc. in the domain of birds and name of a sport, number of players, equipment etc. in sports domain are identified as entities.

Ontology provides a well-defined framework that defines significant concepts (entities) and their semantic relationships [46]. Entities are the basic elements which conceptualize the environment or the domain symbolized by the ontology. In ontology all the relations and properties are defined for entities. Entities are bound to some other entities by relation and to some components within their purview by properties. Then a whole domain is modeled by domain specific entities and definitions of the existence of them within the domain. Fig.2.1 shows a fraction of ontology expressed in OWL which contains examples of entities “*Bird*”, “*Location*” and entity instances “*Penguin*”, “*Mexico*” to illustrate the existence of entities in ontology.

```
<owl:Class rdf:ID="Penguin">
  <rdfs:subClassOf rdf:resource="#Bird"/>
  .....
</owl:Class>

<owl:Class rdf:ID="Mexico">
  <rdfs::subClassOf rdf:resource="#Location"/>
  .....
</owl:Class>
```

Fig 2.1 Fraction of ontology for two Entities *Bird* and *Location* in the domain “Bird”

Finding the entity classes is the key task in ontology construction. Populating ontology with other information is based on found entities. [Appendix A]

2.3.2. Relations in Ontology

As mentioned in the first chapter, ontologies are composed at least of entity classes and relations. Entities are related either taxonomically or non-taxonomically. Taxonomical relations are IS-A relations which exist between an entity class and an entity instance or a class and a sub class. Non-taxonomical relations mostly exists between two entity classes or between an entity class and an object attribute and can also be termed as class or object properties on certain occasions. As shown in the chapter 1 a major portion of the research

work carried out on information extraction for ontology construction, concentrates on taxonomical relations. According to the work done by the various researchers, relation extraction can be viewed in three prospects as follows.

1. Extraction of taxonomical relations

- Numerous methodologies mentioned in the Chapter 1 are in use of successful extraction of taxonomical relations. But those techniques have confined to taxonomical relations only and do not make any provisions for any other relation types.

2. Extraction of pre-defined relations

- Extracting entity classes for pre-defined relations such as *part_of*, *is_occupied* etc. can be considered as an extension to the taxonomical relations. Any other relations out of the pre-defined set of relations cannot be identified. Therefore the relation extraction is restricted to only few relations. On the other hand it is not always practical to categorize relationships into few groups because natural language is enriched with a vast vocabulary and numerous sentence structures which embed various types of semantic relationships. Recent development of using Freebase or an existing database of entities and relations can be considered as an extension to the pre-defined relations although Freebase provides much larger number of relations.

3. Extraction of non-taxonomical relations

- Relation extraction based on the verb in a sentence which involves entity classes, addresses an almost any possible relation existing between the entities. Extracting the main verb constitution from a sentence requires natural language processing techniques to be employed. But the complicated nature of natural language sentences demands numerous sophisticated techniques to be investigated to analyze sentences syntactically and semantically. In the same time successful identification of the verb constituent between two entity classes can cover any possible relation existing in between them. Apart from identifying verbs and defining relations, statistical approaches are also used to induce relations from text documents.

As it was mentioned in the Chapter 1 many researchers in the field have focused on above mentioned 1 and 2 approaches. Although these two types; taxonomical and

predefined relations are sufficient in order to construct an ontology, the ontology cannot be successfully populated without considering all the relations existing between entities.

2.3.3 Common methodologies used in entity and relation extraction

Entity extraction may be applied from small structured text snippets to large unstructured heterogeneous information sources. While a simple text segmentation technique can be applied on small text snippets today's researchers face the challenge of seeking more advanced refined techniques to apply on heavily unstructured information sources.

Rule based, machine learning and statistical approaches are commonly used information extraction methodologies [1, 3, 5, 7, 8, 11, 15, 19, 35, 38, 50, 57]. Since the challenging task of information extraction is to identify the relevant piece of information from unstructured natural text, natural language processing techniques also play a significant role in entity and relation extraction. In addition manually constructed descriptions are wrapped in some information extraction systems [44] which are applied on small and rather structured text snippets. The proposed information extraction methodology is based on natural language parsing and rule based techniques. Rule based techniques are very effective and other techniques can easily be incorporated in rule based systems. Even in commercial environment rule based techniques are widely applicable [84].

2.4. Natural Language Processing

Entities and relations required for ontology construction are normally wrapped in natural language sentences. Therefore information extraction by default involves identifying entities and the verb which binds entities, from a sentence. As such the analysis of natural language sentences syntactically and semantically plays an important role and can become a necessary preprocessing step in successful information extraction. However natural language processing requires heavy linguistic analysis and sophisticated methodologies to convert information wrapped in natural language into a formal language. Parsing has become a first step in natural language processing and it categorizes the lexical terms into syntactic constituents [106]. Semantic ambiguity in natural language is added to the complexity of language processing and the processing techniques cannot expect a success without addressing those issues.

2.4.1 Complexity of Natural Language Text

Although every natural sentence in English contains basic lexicons such as subject noun, verb, object noun etc. they come in various forms and can be separated into numerous sentence structures that makes analyzing a sentence a complicated process. Generally in natural language processing a text is analyzed syntactically and semantically.

In syntactic analysis a sentence is parsed into a valid grammatical structure of the natural language and each word is categorized into a known lexical group. The complicated nature of the natural language text does not permit parsing the entire text into a set of predefined sentence structures and no human is possibly capable of predefining all the valid syntactic patterns for natural language sentences. Different sentences are identified according to the grammar rules relevant to the sentence structure. One of the most difficult issues in natural language processing is to establish grammar rules relevant to all the sentence structures because no one can pre-list all the sentence structures in any language. Semantic analysis involves learning semantics for identified lexical categories in syntactic analysis.

Some sentences are very expressive, but contain very little information. Some sentences are short and appear less complicated, but rich in information.

For an example the sentence which displays the natural language characteristic crossing dependency *Netball is a ball sport played between two teams of seven players* is comparatively short, but contains three pieces of information: *Netball is a ball sport; Netball is played between two teams; Netball team has seven players.*

Therefore a natural language processing system should accommodate uncommon unknown language structures while attempts are being made to fit a sentence to a known structure. But the more complicated and uncommon the sentence structure is, the more difficult the derivation of the information becomes.

2.4.2 Natural Language Parsing and Dependencies.

The Stanford parser [69] is one of the few language parsers available, which not only parse a given sentence to give the grammar rules by identifying syntactic categories, but also give dependencies among linguistic constituents of the sentence. Newest version of the Stanford parser gives the universal dependencies while the previous version produces

typed dependencies. It uses Part of Speech tags provided by the Hepple Tagger for annotating the sentences with syntactic categories. The definitions of Part of Speech tags used in Hepple Tagger are given in the Appendix E.

The Stanford typed dependencies [17] representation was designed to provide a simple description of the grammatical relationships in a sentence that can easily be understood and effectively used by people who want to extract textual relations, without linguistic expertise. The definitions of Stanford Typed dependencies are given in Appendix F. Stanford dependencies are binary relations held between a governor (also known as a regent or a head) and a dependent. It represents all sentence relationships uniformly as typed dependency relations. The governors and dependents are the words in the sentence represented in the relation with their positions indicated by a number. Current Stanford dependencies represent approximately 52 grammatical relations. These dependencies are quite effective in relation extraction. Current Stanford parser gives universal and enhanced universal dependencies. In simple universal dependency representation each word (except the head word of the sentence) in a sentence is the dependent of one other word. In the enhanced universal representation, dependencies involving prepositions, conjuncts, as well as information about the relative clauses are collapsed to get direct dependencies between the context words. This enhancement is often useful in simplifying patterns in relation extraction applications.

2.5 Rule based Techniques

Entity and relation extractions can be conveniently performed through a collection of rules formulated mainly by learnt examples or manual descriptions by a domain expert [15, 40]. Rules which embed manual descriptions are not always applicable to large unstructured data sources where the existence of entities is unpredictable. Therefore the real challenge of rule formulation is to learn rules from labeled examples known as trained data. A typical extraction rule consists of a contextual pattern as a condition at the antecedent and an action at the consequence of the rule (i.e. Pattern \rightarrow Action). In entity extraction contextual patterns consist of one or more labels capturing various features of one or more entities and the context in which they appear.

A contextual pattern is a regular expression defined over features of tokens in the text. The action part of the rule is used to denote labels or tags to a sequence of tokens.

Rules are used in three ways in entity extraction [14.]

(i) For single entity extraction

A typical rule is fired to extract a single entity. In identifying a single entity the rule uses an optional pattern to capture the context before the beginning of the entity, a pattern to match the tokens in the entity and an optional pattern to capture the context after the end of the entity.

(ii) For making the boundaries of an entity.

For longer entities marking the boundaries is more appropriate. Therefore separate rules are defined to mark the start and end of an entity boundary. These rules are fired independently and the tokens in between boundaries can be picked as an entity.

(iii) For multiple entity extraction

Some rules can contain regular expressions with multiple slots, each representing a different entity that results the recognition of multiple entities simultaneously.

2.5.1 Rule learning algorithms

Although some rules are manually constructed most rules are formed extensively by the use of learning techniques. Learning is normally classified as supervised learning or unsupervised learning [102, 103]. Semi supervised and distant supervised methods are also developed in between. Both types of learning techniques learn extraction rules from an available data set. In supervised learning a large amount of data is labeled with data types and known as training data that is the input to the learning algorithm. The learning algorithm is expected to identify patterns in training data and induces data extraction rules incorporating them. In semi supervised learning a small amount of labeled data is used to initiate the learning process and it is then built on the initial data set and the newly found information by the system itself. Distant supervised methods build the learning process based on existing knowledge bases. In unsupervised learning a learning algorithm is used to find hidden structures in unlabeled data.

The body of the rule is expected to cover a number of elements in the available data set. This is called the coverage of the rule. Of all the data elements rule covers, the action

specified by the rule will be correct only for some data items. Then the precision of the rule is defined as follows [105].

$$\text{Precision} = \frac{\text{Number of data items covered accurately by the rule}}{\text{Total number of data items covered by the rule}}$$

The goal of rule learning is to find the rules to cover all the data items in the list with a higher precision. The main purpose of learning algorithms is to form rules to cover all the data in the training set and refine the formed rule set to prevent obtaining the incorrect results. In supervised learning the major principle used in many learning algorithms is to construct the first rule from a seed labeled example and continue rule formation by removing the covered examples from the set until the set is empty [34]

2.5.2 Markov Logic Network for Statistical Relation Extraction

Each rule can have an associated weight that reflects how strong the constraint that it impose. Rules can be modeled in Markov Logic Network (MLN) [52] environment in order to find the weights for the rules. MLN combines first order logic with probabilistic model. A Markov network (also known as Markov Random Fields) is a model for the joint distribution of a set of variables $X = (X_1, X_2, \dots, X_n) \in X$. It is composed of an undirected graph G and set of potential functions ϕ_k . The graph has a node for each variable and the model has a potential function for each clique in the graph. A potential function is a non-negative real valued function of the state of the corresponding clique [52]. MLN requires grounding all the first order clauses by substituting constants for all the variables in them.

The joint distribution represented by a Markov network [52] is given by

$$P(X = x) = \frac{1}{Z} \prod_k \phi_k(x_{\{k\}}) \quad (2.1)$$

Where $x_{\{k\}}$ is the state of the k^{th} clique Z , known as the partition function is given by $Z = \sum_{x \in X} \prod_k \phi_k(x_{\{k\}})$. Markov networks are often conveniently represented as log-linear models, with each clique potential replaced by an exponentiated weighted sum of features of the state, leading to

$$P(X = x) = \frac{1}{Z} \exp\left(\sum_j w_j f_j(x)\right) \quad (2.2)$$

$$Z = \sum_{x \in X} \exp(\sum_j w_j f_j(x)) \quad (2.3)$$

A feature may be a real valued function of the state and its weight is $\log \phi_k(x_{\{k\}})$. Features can be learned from data as conjunctions of atomic formulas.

A Markov Logic Network L is set of pairs (F_i, w_i) where F_i is a formula in first-order logic and w_i is a real number. Together with finite set of constants $C = \{c_1, c_2, \dots, c_{|C|}\}$, which are used to ground the atomic formulas it defines a Markov network $M_{L,C}$ as follows.

1. $M_{L,C}$ contains one binary node for each possible grounding of each predicate appearing in L . The value of the node is 1 if the ground atom is true and 0 otherwise.
2. $M_{L,C}$ contains one feature for each possible grounding of each formula F_i in L . The value of this feature is 1 if the ground formula is true and 0 otherwise. The weight of the feature is w_i associated with F_i in L .

An MLN can be viewed as a template for constructing Markov networks. Given different set of constants it will produce different networks and these may be in various sizes, but all will have certain regularities in structure and parameters given by MLN. From equations (2.2) and (2.3) and the Definition, the probability distribution over possible worlds x specified by the ground Markov network $M_{L,C}$ is given by (2.4)

$$P(X = x) = \frac{1}{Z} \exp\left(\sum_i w_i n_i(x)\right) = \frac{1}{Z} \prod_i \phi_i(x_{\{i\}})^{n_i(x)} \quad (2.4)$$

Where $n_i(x)$ is the number of true groundings of F_i in x , $x_{\{i\}}$ is the state (truth values) of the atoms appearing in F_i , and $\phi_i(x_{\{i\}}) = e^{w_i}$

The probability of a formula F_1 with the given evidence F_2 is computed on MLN by (2.5)

$$P(F_1 | F_2, L, C) = \frac{\sum_{x \in X_{F_1} \cap X_{F_2}} P(X = x | M_{L,C})}{\sum_{x \in X_{F_2}} P(X = x | M_{L,C})} \quad (2.5)$$

Where L is the MLN, C is the set of constants and X_{F_1} is the set of states that F_1 holds and X_{F_2} is the set of states that F_2 holds.

Weights of first order formula can be learnt generatively or discriminatively. Weights can be calculated generatively by maximizing a likelihood or pseudo-likelihood of a relational database. Since the computations in generative learning is highly intractable

and as in many applications as in system explained in here and the pseudo-likelihood parameters may lead to poor results when inference across non-neighbouring variables is required, discriminative learning [85] is preferred in weight learning for relation-extraction-rules. In addition to that a priori which predicates will be evidence and which will be queried is known, makes discriminative learning [85] more suitable for the purpose. In discriminative learning conditional likelihood of query atoms is used. The conditional likelihood of query atoms y given evidence atoms x is given by (2.6)

$$P(y | x) = (1 / Z_x) \exp(\sum_{i \in F_y} w_i n_i(x, y)) \quad (2.6)$$

Where F_y is the set of all MLN clauses with at least one grounding involving a query atom and $n(x, y)$ is the number of true groundings of the i^{th} clause involving query atoms. The gradient of the Conditional log-likelihood is given by (2.7)

$$\begin{aligned} \partial / \partial w_i (\log P_W(y | x)) &= n_i(x, y) - \sum_{y'} P_W(y' | x) n_i(x, y') \\ \partial / \partial w_i (\log P_W(y | x)) &= n_i(x, y) - E_W[n_i(x, y)] \end{aligned} \quad (2.7)$$

Although the number of grounded atoms can be reduced as explained above, computing expected counts E_W is intractable. Closed World Assumption cannot be used with the dependency literals because the domain is infinite though limited number of training data is used in the experiment. Therefore E_W can be approximated by the counts $n_i(x, y_W^*)$ in the MAP(Maximum A Posteriori) state. In the problem domain given under experimental results, finding single MAP state is not guaranteed because same conditional probability value exists for number of states. Therefore Contrastive Divergence (CD) [86] is used in gradient calculations instead of using MAP state. CD approximates the expectations from a small number of Monte Carlo Markov Chain (MCMC) samples. Gibbs sampling is chosen with CD in order to create samples of states. In using Gibbs sampling random numbers are used in assigning truth values for atoms from conditional probability. The conditional probability of each ground atom within its Markov Blanket is used for Gibbs sampling. Each Gibbs step consists of sampling a ground atom when its Markov blanket is given. Gibbs sampling requires weights of rules in its sampling process. The weight of a rule is calculated basically for Gibbs sampling by the log odds between a world where the rule is true and a world where the rule is false when other things are equal. But this

phenomenon cannot be applied to find the actual weight of a rule because rules in system share variables with each other. However the weights calculated in this manner is used only for the sampling. Weight is calculated for each Markov Blanket separately. Then the probability of a ground atom X_i with respect to a Markov Blanket B_i is given by

$$p(X_i = x_i | B_i = b_i) = \frac{\exp(\sum_{f_i \in F_i} w_i f_i(X_i = x_i, B_i = b_i))}{\exp(\sum_{f_i \in F_i} w_i f_i(X_i = 0, B_i = b_i)) + \sum_{f_i \in F_i} w_i f_i(X_i = 1, B_i = b_i))} \quad (2.8)$$

2.5.3 GATE's Information Extraction System ANNIE

General Architecture for Text Engineering (GATE)[14] is an extensive framework and graphical development environment which enables users to develop language processing components in their system, introduced by H. Cunningham, D Maynard and the team in University of Sheffield. GATE provides facilities for information extraction, ontology construction, natural language analysis etc. Borislav Popav et al. [37] has used GATE to develop rules to perform name entity recognition to recognize names with respect to a given ontology. A pre populated knowledge base is also maintained for this purpose and this knowledge base is continuously populated with the extracted entities. GATE Developer is the GATE's integrated development environment for language processing components bundled with a very widely used Information Extraction System (ANNIE) and a comprehensive set of other plug-ins. GATE also provides a framework *GATE Embedded* which facilitates the inclusion of GATE in diverse applications.

GATE contains three main components.

- Language Resources (LRs) representing entities such as lexicons, single document, corpora or ontologies.
- Processing Resources (PRs) representing entities such as parsers, generators or ngram modelers.
- Visual Resources (VRs) representing visualization and editing components that can be used in Graphical User Interface.

GATE allows developers to expand the facilities by building additional resources through its convenient graphical environment. The developer can use the graphical environment and the framework to construct the resources of the above mentioned three types. Therefore when an appropriate set of resources have been developed for various

applications they can easily be plugged into GATE framework. GATE supports documents in a variety of formats including XML, RTF, email, HTML, SGML and plain text etc.

ANNIE an information extraction system included in GATE is a series of processing resources which use finite state techniques and the JAPE language to implement various tasks from tokenization to semantic tagging or verb phrase chunking.

JAPE provides finite state transduction in annotation based on regular expressions. Regular expressions can be recognized by JAPE in annotations on documents. The grammar used in JAPE defines grammar rules, antecedent of which consists of a set of patterns that may contain regular expression. The consequent is the annotation manipulation statements. Patterns matched on the antecedent are referred by the consequent to annotate the pattern elements in the document.

JAPE [14] rules accommodate four features of tokens. Features associated with a token is one or of the followings.

- A string representing the token. e.g. `Token.string == "early"`
- Orthographic type of token that defines the type of the token such as uppercase word, lowercase word, mixed case word, number, special symbol, space, punctuation etc.
e.g. `Token.kind == number`
- The Part of speech of the token e.g. `Token.category == DT`
- Annotation attached by earlier processing step.
e.g. `{Unknown.kind == PN}`
`):loc`

2.6 Ontological Information Extraction for Document Classification.

With the exponential growth of web resources and the higher dimensionality of documents makes the automated text classification task a huge challenge for data mining research community. Assigning text documents to predefined classes of documents online, is considered as an effective approach of finding useful information from numerous online text repositories. In document classification, first documents should be represented by selected features in a way that they can be categorized into respective groups. Most of text classification methods use word counting, word vs. document

proportion as features to represent the document. In general, a document is mainly represented by the concept of bag of words where set of words together with their frequencies are used to represent the document or as a string where the document is represented by sequence of words. In addition techniques such as rough set, principle component analysis etc. [1] are applied in order to find minimum set of features without significant loss of information. These techniques require a considerable amount of effort in finding the most relevant features from a document to be used in the selected text classification method and mostly end up with large number of terms with noisy irrelevant features.

Ontologies can be used to annotate documents with semantic tags as well as annotated documents are needed for ontological information extraction in supervised learning methodologies. . When a vast collection of documents in various domains are tagged with domain specific information they can be easily separated into their respective domains. Therefore ontological information can become a good source for features to use in document classification. Then the same technique can be used to classify document and extract ontological information simultaneously. In using domain specific ontological information for document classification a domain is considered as a class. Classification within a domain class can make the information readily available and can enrich the ontology with additional information..

Chapter 3

Related Work

3.1 Introduction

As mentioned before numerous works has been carried out for entity extraction for ontology development. But relation extraction has not been addressed to such an extensive level. A summary of the work carried in this context is presented in this chapter from a critical point of view. Literature survey on information extraction is grouped according to main technique used in each research work although most work are hybridized with more than one technique. The mainly used methodologies are generation of rules, natural language processing, statistical methods and machine learning approaches. Since the methodology developed for ontological information extraction in the research work presented in this thesis is adapted to use for document classification as proof of concept, a literature review on text classification is also included in this chapter.

Section 3.2 focus on literature of the rule based methodologies. But these rule based research work also use natural language processing, machine learning and statistical methods in generating extraction rules.

Natural language processing based work is discussed in section 3.3. Some of the systems presented in this section incorporate extraction rules too.

Section 3.4 includes statistical approaches employed on information extraction. Machine learning techniques and tools are used in many of these researches.

Section 3.5 presents a review on state-of-art methods and some newly developed methods on text classification.

Summary of the related work is given in the tabular form in Appendix B.

Section 3.6 highlights the common problems and weaknesses of the previous systems, identified from the literature review.

3.2 Using Rules in Information Extraction

Some systems that have been developed for ontology construction/population can extract entities only while some systems can extract both entities and relations [11, 16, 10, 8, 46, 7, 15, 35, 36, 48]. Many supervised learning systems induce extraction rules based on

identified patterns in the training data set. In identifying patterns the learning algorithms use the features of labeled data and their neighborhood words. Therefore the successful application of extraction rules depends on the identification of appropriate neighbourhood and features of the language tokens. Research on generating extraction rules mainly focus on two contexts: creating wrappers for the web and developing general purpose information extractions systems for natural language text [3, 5, 7, 10, 16, 19, 46, 44, 48, 50].

Wrappers contain a set of extraction rules suitable to extract information from a web site. Two systems developed by Craig A. Knoblock et al [12, 13] enable constructing wrappers for web based information sources. Both systems use machine learning to induce rules for wrappers and also to produce answers for user queries. One system named Ariadne [13] is capable of integrating web sites to gather information for user queries. But, many training examples and heavy user involvement are needed to provide data to induce extraction rules. For the other system, Knoblock and the team have developed STALKER, [12] a wrapper induction algorithm that learns extraction rules based on examples labeled by the user. Wrapper induction systems normally make use of the HTML structure of the web pages. This system requires only a smaller number of examples and it has demonstrated its applicability on short web pages of a similar structure. DIScoTEX [36] integrates information extraction and knowledge discovery from databases(KDD) to fill values for the slots in a template of a particular entity. In DIScoTEX RAPIER [30] and BWI (Boosted Wrapper Induction) [31] algorithms are used in information extraction to construct a database from a set of documents. KDD techniques such as RIPPER [31] and APRIORI [88] are used to induce association rules based on the database in order to find additional facts to confirm the extracted information and to find interesting relationships. There are no weight assignments for rules. DIScoTEX uses manually constructed dictionary to handle synonyms. The test results show that the system performs efficiently on short web pages of similar structure. (E.g. Job Posting, University web pages etc.). RAPIER [30] can't distinguish different occurrences of the same term. When a concept is associated to a particular slot in the template, that concept will always be associated with that slot. Therefore, identifying

different relationships of an extracted concept is not possible. Wrappers perform better on short structured or semi structured web pages.

Amilcare [8] is a system which can only extract entities. It provides an annotation tool and an information extraction tool and the user can use both or just the annotation tool as an assistance to the manual annotation. Amilcare uses (LP)² [9] algorithm to generate extraction rules from a training corpus. It also exploits natural language processing techniques. LazyNLP [9] is used on the training corpus annotated with XML tags to include linguistic information. (LP)² induces two types of rules; tagging rule that insert annotations in the text and correction rules that corrects mistakes in the annotations. Then the information extraction tool silently works in the background and suggests further annotations based on the annotations made by the user. But in this system, extraction rules which are not weighed, can be used only to extract entities and are demonstrated on the short texts such as announcements. According to Amilcare developers, this information extraction process integrates into the user's usual working environment without requiring any preparation from the user.

Ont-O-Mat [24] is a system which plugs Amilcare into it as its information extraction component. Ont-O-Mat is the implementation of a tool known as CREAM [24] and develops S-CREAM [23] with Amilcare as the information extraction component. In CREAM an IE component is not built in to Ont-O-Mat; but few components are included to gather information and to build a local knowledge base for text annotation. Further, Amilcare is integrated in to MnM [47] which provides an interface to select a predefined ontology, for information extraction. Then MnM can be considered as a front end of Amilcare. Since relation extraction is not performed by Amilcare, work has been continued by Amilcare team resulting T-rex [30]. T-rex has been developed to incorporate relation extraction into the system. It is a test bed for experimenting with extraction algorithms and extraction scenarios. T-rex's modular architecture implements plug-ins to its components; processor, classifier and combiner. Processing component which is a natural language processing dependent, is composed of processors and feature extractors and classification component which is machine learning dependent composed of classifiers and feature selection algorithms. T-rex [30] features canonical graph based data model [31] to be used by the algorithms. T-rex's data model allows expressing

various links such as grammar link, links related to HTML tags etc. in the document and promote rapid prototyping of new algorithms. Therefore T-rex can be considered as a tool to be used for the implementation of relation extraction scenarios.

Burcu Yildiz and Silvia Miksch [50]'s approach use ontology model to generate rules in order to extract information from text corpus complying with the entities, instances and properties in the ontology. Therefore the system can only extract instances for the sub-classes and values for the data_type property in the ontology. But they have addressed the issue of adapting their information extraction system in different domains. They have incorporated an ontology management module to tackle different domain ontology to serve this purpose. The system is not capable of identifying Non-hierarchical relationships. On the other hand the presence of domain ontology is essential for rule generation module to generate extraction rules which are not assigned any weights.

Atiken [1] uses the Foil algorithm [39] to learn attribute value relations from sentences marked up with relations in the domain. Atiken's ontology based approach is focused on very specific domain; global warming. Background theory to construct rules contains predicates clauses from the text and semantic theory. There is no weight learning process for the rules. Success of the system depends on the selected training sentences .

Drumond et al [51] preprocess text to filter the noun and noun phrases and extract them as terms based on tf-idf measure. Then the extracted terms are wrapped in first order logic and a set of rules are modeled in Markov Logic Network (MLN) [52]. Three hand coded rules infer a term into a concept and one of those rules uses language dependency. In addition to that two more general rules are used. Markov logic probability is used in assigning a term to a concept. This method cluster textural surface forms that has similar meaning using latent relation models. The clustering does not provide reliable implications in generalization of clusters. OntoShopie [5] is a system which generates extraction rules for concept node definitions in order to populate ontology. Documents are manually annotated with predefined tags by the system based on an existing ontology. OntoShopie facilitate this annotation process with a tool which offers the user the set of predefined tags to annotate the document. The documents are then preprocessed by the Natural Language Processing system Marmot. The dictionary induction system Crystal [45] is used on annotated preprocessed documents to generate

extraction rules for concept node definitions. Badger (the Information Extraction component of Ontoshopie) extract values from the concept node definitions to fill slots with possible values, in a frame based template. OntoShopie addresses the issue of selecting best rules by assigning a confidence value for each rule. System has being constructed based on a very specific domain which involves rather small documents of research projects carried out in a university and evaluated on the same domain. Entities in OntoShopie are considered as events and, events should be manually defined with their properties. These events and their properties are the guidance for the predefined tags offered for the user to annotate documents. Since the system is not customized, working in different domain requires heavy user involvement in defining the events and their properties for the input document corpus.

3.3 Natural Language Processing based Techniques

Use of natural language processing techniques to preprocess the information source has become an essential approach to boost the ontological information extraction process whether it uses extraction rules or not. Some of the work discussed in this section use extraction rules incorporated with natural language processing as Amilcare system which is discussed above. Parsing natural language text and identifying syntactic constituent of sentences (Part of Speech tagging) catalyze the whole process.

Roxana Danger and Rafeal Berlanga's work [15] concentrates on extracting entity instances from a parsed natural text, using OWL [93] ontology. They use a similarity function between text fragments and lexical description in the ontology to extract entity instances. Several inference rules in the ontology and segment scope definitions that indicates which other segments can be related to a text fragment are applied to add new relations to connect instances. Since extracting instances for entities in an already existing ontology is the prime concern, the presence of ontology is essential for the system to be invoked. The system does not create ontology, but populate them.

The system developed by Nadzeya Kiyavitskaya et al. [33] adapts a methodology from LS/2000 software analysis [18] to mark up the documents with XML grammar. Their approach was to do shallow parsing of the documents and use structural and lexical patterns to identify basic entities such as time, date, money etc.

The ultimate goal of natural language processing is to form a logical interpretation of natural language text. Hoifung Poon and Pedro Domingos [38] propose OntoUSP, a system that learns hierarchical relations over clusters of logical expressions and populates them by translating sentences to logical form which is a promising initiative as it relaxes the complexity of natural language.

Five syntactic patterns (i.e. lexical patterns like such as, including etc.) introduced by Marti A. Hearst [25] for taxonomic relations are applied in many systems for relation extraction. OntoSyphon [19] extracts instances for primitive classes identified in an ontology using five Hearst phrase [25] template with an associated learning accuracy figure which depends on the number of time an instance appear with a class in one or more Hearst phrases. OntoSyphon can be adapted to different domains to extract such information. As the system is restricted to search for only five phrases it will miss any information which is not fallen into the vicinity of those five phrases. Associated learning figure is used to filter out irrelevant instances. However a huge document collection is required to compute the learning accuracy for an extracted instance as it is based on frequency of the instance appearing with one or more Hearst phrases. There is no possibility of using word specific features(tokens) with OntoSyphon. There is room to improve overall performance by incorporating other techniques such as domain-specific pattern learning, combination of multiple sources of evidence. At present only taxonomical relations can be dealt by OntoSyphon. Text2Onto [7] developers also have developed JAPE[14] rules which is an integration to GATE framework, to implement matching Hearst phases for identifying concepts and instances [7]. The novelty of the Text2onto system is the Probabilistic Ontology Model (POM). Here, extracted information stored in the POM, gets assigned a value indicating how certain the algorithm used about the existence of the corresponding instance and can be translated later to any ontology language construct. POM assists not only to filter the irrelevant information, but also to detect differences of a particular document corpus to a previously used corpus and incorporate the differences into ontology. Text2Onto can identify taxonomic relations, mereological (part-of) relations and some general relations of restricted sentence constructs. The system employs a shallow parsing strategy to extract few syntactic frames and map these frames to ontological relations. Text2Onto can

extract the syntactic frames for; transitive verb, intransitive verb + PP-complement(Propositional Phrase) and transitive verb + PP-complement only. It can't handle syntactic phrases out of the above mentioned frames. Diana Maynard, Adam Funk and Wim Peters [35] have also investigated three linguistic patterns including Hearst patterns for the development of the tool SPRAT [35] in GATE to extract variety of entity types and relations between them. .

Armadillo [10] is a system for producing automatic domain-specific annotation on large repositories in a largely unsupervised way. The system is initiated with a seed lexicon (probably an entity drawn from ontology) and uses regularities in web pages to learn rules for wrappers which are pieces of software that enables semi structured web sources to be queried as if it were a database. It exploits the available tools (e.g.NER) for information extraction to facilitate the rule learning. Extracted information is confirmed using other web services and is stored in RDF [95] store as Subject-Verb-Object triplets for document annotation. Annotated portion of the document is used to train more sophisticated information extraction (IE) engines. Subject-Verb-Object triplets are used to identify the relations. Armadillo does not consider unannotated text as negative examples as many others [5, 8, 9, 21, 22, 49] machine learning IE systems do. Unannotated portion of the text is further processed using rules learned to find out more entities. Another advantage of the system is that it can address issues of some syntactic and semantic ambiguities because it integrates some other web services to confirm the accuracy of extracted information. According to the results shown by the application of Amadillo on three domains; Computer Science, Art and Geography, it can easily be switched to various domains and it is very effective for extracting basic terms such as names from a document corpus and cannot expect to be effective on non-basic terms.

R. J Mooney [36] uses ILP technique to extract relational patterns from natural language data. They try to discover rules for Link Discovery which concerns the identification of complex relational patterns that indicate potentially threatening activities in large amounts of relational data. Their approach is completely for domain specific task. OntoMiner [16] uses semantic partitioning [16] for extraction of concepts, instances and taxonomic relations in order to develop ontology. Frequently occurring terms in a XML document corpus are extracted as concepts. In this case the HTML pages are converted to

XML documents using flat partitioning. Hierarchical partitioning is used to identify direct or indirect parent-child relationships in order to extract taxonomies. Instances are identified using the links in a document. It can identify lexicographically related terms for e.g. *book* and *books*, but not semantically related terms for an example *world* and *international*.

Open Information Extraction systems (IE) are employed to capture all types of relations present in text documents. Open IE systems extract relational tuples from text without requiring predefined vocabulary, by identifying relation phrases and associated arguments in arbitrary sentences [53]. REVERB [54, 55] and WOE [56] are such state-of-the-art open IE systems. OLLIE [53] is an improved open IE system developed to address the weaknesses of REVERB and WOE. OLLIE learns open patterns to extract relations embedded in various sentence structures of the high precision seed tuples obtained from REVERB. The patterns include dependency path between two arguments which are bound by the relational language construct. The present work showcased by this thesis also uses dependency path information in order to generate relation extraction rules. Carlson et al [57] use seed instances and patterns (seed instances specified by human and patterns according to generalized hearst) on a large corpus to extract more instances and patterns for concepts and relations. Some constraints are coupled with extractors to enhance the accuracy of extracted information. Constraints are imposed on mutual exclusion, relation argument type and multi view of text features.

Recent development [58] in information finding and extraction uses large knowledge bases such as Freebase to link the text with the concepts in the knowledge base. Significant progress has been made in learning semantic parsers for such knowledge bases specially for Freebase. These methods are typically evaluated on question answering tasks and are designed to only parse questions which are completely supported by ontology. Choi et al [58] present a new semantic parsing model and semi supervised learning approach for reasoning with partial ontological support. The method reportedly demonstrates strong performance on entity attribute extraction and will be effective on highly subjective entities. Novel approach in Mints et al's work [59] is the distance supervision method. In their method they have relations in freebase to extract lexical and syntactic features for feature vectors of their multiclass logistic classifier. They use

dependency path as their syntactic feature which along shows promising results according to their evaluation. Yao et al [60] perform weak supervision while using selectional preference constraints to jointly reason about entity types. Here in place of annotated text, only an existing knowledge base is needed to train a relation extractor. The facts in the KB are heuristically aligned to an unlabeled training corpus and the resulting alignment is the basis for learning the extractor. Naturally the predictions of distant supervised methods are subjective depending on the availability of domain specific information in the knowledge base.

3.4 Statistical Methods for Ontological Information Extraction

In using statistical methods the text is categorized into tokens or word chunks. Statistical techniques are used to assign labels for these tokens or word chunks based on predefined entity extraction features such as word features, orthographic features, dictionary look up features etc. A typical extraction task depends on a diverse set of clues capturing various properties of the token and the context in which it lies. Hidden Markov Models [99], Conditional Random Fields and Support Vector Mechanism [98] are some of the most prominently used statistical models in information extraction.

PubMiner [21] which has been developed to extract entities and relationships from massive biological literature uses Name Entity Recognizer and natural language processing to identify entities. It uses a Part Of Speech (POS) tagger based on Hidden Markov Model for tagging biological words as well as general words. The named entity tagger based on support vector machines recognizes a region of an entity and assigns a proper class to it. Event extractor which considers a verb as an event finds the binary relation between two name entities identified in the sentence where the verb is extracted. Features of extracted entities and event verbs are identified using public medical databases and the feature set is further refined to filter out unnecessary information. Association Rule Discovery using Apriori algorithm [88] is used to generate rules to predict interaction between entities. Although PubMiner is capable of extracting both entities and relationships, it also extracts many false positives in the domain. In finding relations PubMiner heavily depends on external resources such as public medical database and treasures

Hui Han [22] have concentrated on the headings of research papers to classify header information into 15 classes of entities. They have used Support Vector Machines (SVM) for this purpose and, implemented the system using SVMlight [96]. Lines in the header can contain entities that belong to multi classes as well as to a single class. The system uses line and word specific features to form a feature vector for independent line classification and the feature vector is extended further by concatenating with the class labels of N lines (identified using independent line classification) before and after the current line to improve the line classification task. System performance is extended to extract metadata from multi-class lines by forming patterns to identify chunk boundaries in multi-author lines. Currently they assume that each line has only one chunk for each class. This is not appropriate even though it is rare for a class to have multiple chunks in one line. The entity extraction here is restricted to the headers of research papers and had not addressed the entity extraction from a text.

OntoLT [3] is a plug-in for the widely used Protégé ontology development tool that supports the interactive extraction and/or extension of ontologies from text. OntoLT is based on linguistic analysis to find main entities as head nouns for Protégé classes and other nouns as modifiers for sub classes. Precondition language provided by OntoLT allows user to define mapping rules although a number of mapping rules are predefined and included with OntoLT plug-in. This provides an environment for the user to experiment with more techniques. Mapping rules are used to extract or extend ontology with the linguistically analyzed entities. Statistical preprocessing with chi-square function for determining domain relevance is done to identify domain specific terms with respect to a general corpus. OntoLT heavily relies on linguistic analysis.

WebKb [11], is an extensive work carried out by Carnegie Mellon University in information extraction and text mining at an early stage as 1999 and uses both machine learning and statistical methods. The system has a very wide scope and, is based on domain ontology. It is capable of extracting classes and relations from the web pages and also text fields from unstructured text. Here, relevant web pages are categorized into classes identified with the domain ontology and relations are extracted using the hypertext in the categorized pages. It uses machine learning algorithm Foil [39] to learn classification rules for identifying class instances. A similar algorithm [11] is used for

relation extraction. In addition the system demonstrates the use of statistical methods such as Naïve Bayes [49] for the same purpose. Sequence Rule Validation (SRV) which produces a set of information extraction rules based on labeled pages and a set of features defined over tokens is used for extraction of text fields. The evaluation of the system shows that it efficiently works in categorizing web pages from a single site or a cluster of related pages where there are links between pages for navigation. According to the evaluation of the system it can be seen that it yields satisfactory results within certain limitations such as class instances are restricted to a single web page, only known relations defined in the ontology are identified based on hyperlinks in categorized web pages etc. The developers of the system have demonstrated the extraction rules generated for basic text field extraction. The test results shows that the system suffers from the problem of extraction of negative instances by the rules, though many systems, developed even later are affected by the same. The proposed technique developed for information extraction is used on document classification too. Then some relations can be readily identified through sub classification by the class name somewhat similar to identifying relations through hypertext in the categorized pages as in WebKb.

T. Wang et al [48] has addressed hierarchical relation extraction using SVM based approach. They have experimented on ACE2004 [70] training data which are annotated texts for entity and relations according to ACE program definitions. ACE2004 defines a hierarchy of relation with 7 top types and 22 sub types. These relation types include most commonly used general relationships. SVM models are built for detecting the relations, predicting the type and sub type of relations between every pair of entity instances in a same sentence. Detection of relation is based on the context information within a sentence. SVM models are created by feature vectors generated during the training phase. Features are derived from number of GATE-based language processing tools applied on the training documents.

Yao and Haghighi et al [61] use unsupervised approach for relation extraction between the entities. Their method is based on Latent Dirichlet Allocation (LDA) on topic model. Relations are induced at the documents level. Relations are modeled from a selection of relation distribution which has a dirichet prior. They also exploit features on the

dependency path between entities. Evaluation results show the recall and the precision of frequent relations extracted from a corpus of new paper articles.

Pawar et al [62] have used three maximum entropy classifiers based on entity features to predict entity types and relation types. Two classifiers local entity classifier and local relation classifier do the predictions independently. The other classifier, pipeline relation classifier uses the output of the local entity classifier (i.e. entities) in predicting the relation type. Their work is based on ACE 2004 data set and therefore the entity types and relation types are restricted to ACE 2004 entity and relation types except NONE and NULL. Weight learning methods were not used though rules are modeled in Markov logic network. Weights of the rules are determined by log odd ratio and constant multiplier.

In Riedel et al's approach [63] uses latent features of relations and tuples involved in the relations. Their approach is based on extensions of probabilistic models of matrix factorization and collaborative filtering. They present the probabilistic knowledge base as a matrix with entity pairs in the rows and relations in the column. The rows come from running cross document entity resolution across pre-existing structured databases and textual corpora. The columns come from the union of surface forms and DB relations. They claim that their model can predict relations which do not exist in the Freebase.

3.5 Related work on Text Classification

A wide range of text classification methods have been established for various applications. The methods are still been investigated at the direction of strength and weaknesses for the purpose of possible improvements. Here most common widely established document classification techniques along with recent developments are discussed.

The Naive Bayes classifier [64] is the simplest of the models which embodies the strong assumptions about how the data is generated, made by Bayesian probabilistic approaches. These probabilistic approaches use collection of labeled training examples to estimate the parameters of generative models. Classification of new examples is performed with Bayes rule by selecting the class that is most likely to have generated the example. The Naive Bayes classifier assumes that all the associate attributes are independent of each

other given the context of the class. There are two types of Naive Bayes classifications both of which make the Naive Bayes assumption. In both methods the posterior probability of a class for a given document is calculated and the class with highest posterior probability is then assigned to the document.

(i) Multivariate Bernoulli Event Model

In this model a document is represented by a vector of binary attributes indicating which words occur and which words do not occur in the document. The number of times a word occurs in the document is not considered. When calculating the probability of a document one multiplies the probability of all the attribute values including the probability of non-occurrence for word which do not occur in the document.

The posterior probability $P(C^T = i | P(T = Q))$ is needed to find where C^T denotes the class of sampled term set T and T is a sample from the term distribution of class i and Q is the term distribution of the document.

$$P(C^T = i | P(T = Q)) = (P(C^T = i) \cdot P(T = Q | P(C^T = i))) / P(T = Q)$$

$$P(C^T = i | P(T = Q)) = (P(C^T = i) \prod_{t_j \in Q} P(t_j \in C^T = i) \prod_{t_j \notin Q} (1 - P(t_j \in C^T = i))) / P(T = Q)$$

(ii) Multinomial Event Model

A document is represented by a set of word occurrences from the document. As above the order of the word is not considered but the number of times that word occurs in the document is considered. When calculating probability one multiplies the probability of words that occur.

$$P(C^T = i | P(T = [Q, F])) = (P(C^T = i) \cdot P(T = [Q, F] | P(C^T = i))) / P(T = [Q, F]) \text{ Where } F \text{ is the frequency of the term.}$$

Bernoulli model is suitable for short documents. Classification accuracy of NB classifiers is affected by imbalanced class distribution where some classes have more training examples than others and by attribute independence assumption. Strong word dependencies render a bias towards word probability calculations because dependent words often occur together. NB classifiers cannot capture these word dependencies. Therefore word dependency bias is not taken into account in probability calculations. Since the probability calculations are parameterized by class priors the classification accuracy depends on the class distribution of the training document set. Therefore a balanced set of training documents is required to represent each class well in the

distribution. Renny et al [65] introduce a method called complement class formulation to address the effect of imbalanced class distribution on text classification. In their compliment method the parameters are estimated using data from all the classes except the class for which the probability measures are calculated. In some classes, term vectors of training documents contain words which are distributed across the classes and those words also play an important role together with other words in the classification process. In those situations this compliment class method may not give accurate classification. This can also lead to unnecessarily complicated computations when the number of classes in the training corpus is high. Further they propose normalization of word probabilities in order to minimize classification errors occurred due to Naive Bayes independence assumption. Tang B et al [66] propose to use class-specific features with Bayesian classification. Probability Density Functions (PDFs) in the raw data space are reconstructed from the PDFs in low dimensional class-specific feature space according to Baggenstoss's PDF Projection Theorem (PPT) in order to apply class-specific features in Bayesian classification approach.

Support Vector Machines (SVM) plays an important role in text classification [67] and has been widely used in many applications. T.S.Furu et al [68] shows the application of SVM in classification of tissue samples and Drucke et al [69] demonstrates the use of SVM on email data for classifying it as spam or non-spam data. It was shown that the SVM method shown much more robust performance as compared to many other techniques such as boosting decision trees, the rule based RIPPER methods and the Rocchio [70], [71] method. SVM are a form of classifiers which attempt to build good linear separators between classes. Finding the best separator is essentially an optimization problem. This sometimes can be slow especially for high dimensional domain such as text data. In addition generation of good linear classifier is not always guaranteed. When the classification becomes nonlinear the document vectors should be projected to a linear space by a kernel function making the process more complicated.

K-nearest neighbor (KNN) classification finds k nearest neighbours from a training document set for a test document based on the similarity between the test document and a training document. Class candidature is scored based on the classes of neighbours and the class with highest score is assigned for the document [72]. KNN is straightforward and

remarkable classifier which has been shown as one of the most effective methods for text classification [73], [74]. But it suffers from several drawbacks such as sensitivity to skewed class distribution, irrelevant or noisy features which has no exception with KNN also and parameter tuning. Further the success of KNN classifier depends on the availability of effective similarity measures. Generalized instance set (GIS) algorithm [75] introduced by Lam et al uses k nearest neighbours in their document categorization process. In their document categorization process they find a generalized feature vector to represent k-nearest neighbours in a category by Rocchio or Windrow Hoff algorithm [71] and rank the neighbours according to the generalized representation. There are number of generalized instances for a category in the generalized instance set after the generalization process. Then again there is a possibility of losing important category features in the generalization process leading to classification errors and all other weaknesses of the KNN method except effect of skewed class distribution still prevail. Therefore they propose a meta learning method based on multivariate regression analysis to select the most suitable algorithm in generalization for each category in order to minimize the classification error. But using different algorithms for different categories may not be appropriate in some applications.

Centroid [73, 74] is a remarkable classifier in which each class is represented by its centroid and a test document is assigned to the class label by its closest centroid. Centroid combines prevalent features within each class centroid to make it distinctive and separable from others. Although the class centroid of majority classes tends to contain some features of minority classes the average weights of those features in majority classes are much smaller than those in minority classes. Therefore centroid based representation model is less likely to be biased towards majority classes. However Centroid can lead to miss-classification when documents are not linearly separable by the boundaries between class centroid [73].

Pang et al [74] propose a scalable effective flat classifier called CenKNN by combining efficient centroid based text classification techniques. CenKNN has been proposed to improve the effectiveness of KNN on high dimensional and large-scale corpora with imbalanced class distributions and irrelevant or noisy term features. The basic idea of CenKNN is to use an effective and efficient class centroid based dimension reduction

method to substantially reduce the dimensionality of documents and then employ K-D tree structure to conduct a rapid K nearest neighbours search for KNN classification. Their dimensionality reduction method CentroidDR first compute centroid of all the classes and then map documents into the class centroid based space via cosine similarity measure function. CentroidDR reduces the dimensionality of a document representation to number of classes in the training corpus.

In general the rule based systems created for document classifications contain combinations of words taken from the training documents in the condition part of the rule resulting a large number of such combinations. Then the main weaknesses of the rule based systems become the large number of such word combinations and rules in the system. In generating rules Apte et al [76] use an iterative methodology Swap-1 [77] that determines the single best rule related to any particular class. The best single rule achieves the complete predictive value and number of such rules is generated to cover all the training samples with each rule containing much number of components in the antecedent. Therefore the initial set of best single rules is pruned by deleting weak rules and components, allowing an acceptable error rate. Zaiane et al [78] have proposed a method of generating association rules by pruning the number of rules and different terms (terms in the item set) appearing in the condition of the rules, based on support and confidence measures. Haralambous et al [79] use dependencies in a sentence to select the words to include in the item set of the entire document by imposing constraints on dependencies. They further use the WordNet lexical database to replace the words in the item set by the members of their most significant hyperonymic chains.

A classification technique based on information extraction is presented in Riloff et al's work [80]. They present three algorithms which use varying amount of information to classify texts. The relevancy signature algorithm uses linguistic phrases; the augmented relevancy signature algorithm uses phrases and local context; case based text classification algorithm uses larger pieces of context. They explained Relevancy Signature Algorithm as their first attempt to use natural language processing. Relevancy signature is a combination of word and concept node that it triggers and both together represent a linguistic expression. Domain specific dictionary of concept nodes is used to extract relevant information from a sentence to classify texts on the basis of linguistic

expressions instead of isolated keywords. These linguistic expressions are represented as signatures with relevant documents and highly correlated signatures are identified by using statistical techniques. These signatures are used as indices to classify documents. First they used relevance signatures which are short linguistic expressions and then the method is improved by adding more information in the form of slot and filler tuples to augment the relevance signatures because relevance signatures alone lead to misclassification. Due to poor Recall, augmented relevancy signatures are further expanded by adding case based information. In case based method, cases are constructed from each sentence in the document and new cases are compared at the classification phase with thousands of cases already created at training phase in order to find the relevancy of the new cases. However, the constructing cases from sentences in a document is not a good practical approach with lengthy documents.

3.6 Weaknesses and Problems Identified from the Literature

Review

This chapter discussed the previous work on information extraction based on extraction rules, natural language processing and statistical methods. All kinds of learning methods supervised, unsupervised, semi supervised and distant supervised methods are used in the reviewed research work. Limitations and drawbacks some of which may be common to learning method are pointed out from the previous systems. When summarized the reviewed work with respect to the weaknesses the following drawbacks and limitations are mainly identified and majority of them are lined with the research questions put forward in Chapter 1 confirming the need of addressing them further.

- (i) Difficulty in adapting to different domains
- (ii) Low recall values especially in rule based systems and extraction of false positives leading to low precision values
- (iii) Limitations in relation extraction.
- (iv) Insufficient handling of semantic ambiguity in natural language text
- (v) Dependence of performance on the nature of the text documents used (corpus biased performance)
- (vi) Requirement of large amount of training in supervised learning methods

- (vii) Difficulty in mapping the grouped data into an extraction schema in Unsupervised learning methods.
- (viii) Absence of proper weight assignment method for extraction rules.
- (ix) Uncertainty in finding the set of minimum number of document features in classification
- (x) Difficulty in finding proper classification boundaries in linear classification methods.

Attempts are made to address the above mentioned issues in the research except (vii) because (vii) is a weakness of unsupervised information extraction and it does not arise in supervised or semi supervised learning. The above (iv) semantic ambiguity problem is handled only with regards to relation verbs for extraction of relation instances accurately and addressing the issue beyond that point is out of the scope of this research. Weaknesses (ix) and (x) identified in the existing document classification methods can be avoided when the information extraction method presented here is used for document classification.

Chapter 4

Ontological Information Extraction

4.1 Introduction

The technique used for ontological information extraction is discussed in this chapter. Following the identification of entity instances, relation extraction can be performed based on identified entities. As mentioned in chapter 2 GATE [14] includes numerous resources for text processing and annotating information. Therefore it was selected to use for ontological entity extraction.

Relation-extraction-rules are generated based on language dependency constructs. Dependencies produced by Stanford parser are preprocessed to filter required information for rule generation. Relation extraction described here is fulfilled mainly by a two-step process: Rule Learning and Relation extraction by using the learned rules. Rule learning is facilitated by a supervised learning approach and Inductive Logic Programming (ILP) [34] has been chosen as the learning method, for the purpose. Candidature of language dependencies for ILP is explained in this Chapter. Each rule can have an associated weight that reflects how strong the constraint that it impose. Relation-extraction-rules are modeled in Markov Logic Network (MLN) [52] environment in order to find the weights for the rules and to assess the certainty of the extracted relations. MLN combines first order logic with probabilistic model. Weight learning process for relation-extraction-rules in the Markov Logic environment is also described in this chapter.

Sections 4.2 explains how GATE can be used in entity extraction and extend to bundle extraction rules into new processing components related to different domains.

Section 4.3 explains how to represent useful information extracted from natural language in a formal manner and how to deal with semantic ambiguity in order to identify the correct relationship between two entities. It describes the preprocessing of annotated natural language sentences by using language dependency constructs. How relative clauses and conjunctions can be made useful in relation extraction is also explained here.

The Inductive Logic Programming algorithm used in order to generate extraction rules is explained in detail in the Section 4.4. In this Section the application of the adapted

algorithm is shown with examples and certain adverse issues of the algorithm is also addressed.

Section 4.5 explains the applicability of MLN on relation-extraction-rules. The modeling of the rules in MLN is explained extensively with examples. The entire weight learning process for the rules is described in this section.

4.2 Ontological Entity Extraction

In the research GATE is used to identify domain specific entities. Additional plug-ins are needed to extract domain specific entities and they are loaded to the system by the users. Therefore GATE can be expanded as long as the memory of the machine allows. ANNIE already provides general entities such as person's name, location, money etc. More information on ANNIE's capabilities is given in Appendix C. GATE can be enriched with any number of domain specific entities as long as relevant components are plugged into it.

A component has been built in Jape for the domain of birds in order to identify the entities: *Bird*, *Height*, *Length*, *Weight*, *Colour*, *Part*, *Habitat* and *Diet*. The entity "*Location*" can already be identified by ANNIE's Name Entity Recognizer.

4.2.1 Extending GATE's facilities with additional components.

In generating Jape rules gazetteers, [Appendix C] features of tokens themselves and neighborhood features of tokens have been used. The token neighborhood contains eight tokens with four to left and rights each. For examples, for annotating an entity with "*Colour*" and "*Bird*" gazetteers lookup can be directly used, for identifying measurements such as "*Length*", "*Weight*" etc. only characteristics of token entity have to be used and when annotating a document with *Habitat* and *Diet* characteristics of neighborhood tokens are incorporated into annotation pattern description of Jape's rule. When creating a gazetteer for the entity *Bird* both singular and plural terms of a bird name are needed to be included.

For e.g.,

The rule ColorId uses the gazetteer named "colours" to annotate entities with *Colour*, the rule WeightX requires the features of token sequence of the entity for the measurements

entities such as *length*, *weights* etc. and the rule DietBird requires neighborhood features for accurate identification of entity instances for “*Diet*” in the pattern description of the Jape rules for the domain *Bird*. The rules ColourId, WeightX and DietBird are shown below

```
Rule: ColourId
(
  {Lookup.majorType == colours}
)
:color -->
: color.Colour = {rule = "ColourId"}
```

```
Macro: Weigh
({Token.kind == number}
(({SpaceToken}
  {Token.string == "kg"}|
  {Token.string == "Kg"}|
  {Token.string == "g"}|
  {Token.string == "oz"}))
)
```

```
Rule: WeightX
(
  (Weigh)
  :weight
)
-->
:weight.Weight = {kind = "weight", rule = "WeightX"}
```

```
Rule: DietBird

( { Token.string == "diet"}
  { Token.string == "of" }?
  { Token.string == "the"}?
  ( { Token.category == NN }| {Token.category == NNP} ) ?
```

```

{Token.category == VB}?
{Token.category == VBN }?
{Token.string == "of" }? ) |
( ( { Token.string == "eat" } |
    { Token.string == "eating" } ) ) |
( ( { Token.string == "feed" }
    { Token.string == "on" } ))
: diet
)
---→
: diet.Diet = {kind = "diet", rule = DietBird }

```

Since a supervised method has been used an annotated training text corpus was used to identify token and token neighbourhood features. GATE also facilitates the manual annotation of a text corpus and then a processing component can be built into GATE to identify the token and token neighbourhood using the annotated training corpus. A threshold value for feature weight is defined for considering a feature to be used in a JAPE rule for an entity. A feature weight is computed as follows

$$\text{Feature weight} = \frac{\text{Number of annotations for the entity with the feature}}{\text{Total number of annotations for the entity}}$$

JAPE rules are initially built with the features of highest feature weights. Highest priority is assigned to the rule generated first with the features of highest weight. The main problem in using neighbourhood features is extraction of inaccurate entities. Therefore rules are verified by applying them on the training data and modified by augmenting with the counterfactuals which will prevent the extraction of false entities. Since the focus of many researchers in ontological information extraction has been the entities there are provisions to enhance entity extraction further with the available techniques described in Chapter 3.

Some annotations drawn from specific domains can be used in any domain where applicable. For an example above mentioned all three entities are applicable not only in the domain of bird but in other domains also where *Colour*, *Diet* and *Measurement*

entities are present. Fig 4.1 shows the GATE's user interface with a document from the corpus "*Bird*" annotated with the checked entities appeared on the right side of the interface. Program coding for Jape rules is given in Appendix D.

For entity extraction for ontology construction a plug_in has been developed to accommodate more entity rules for GATE.

Fig 4.2 shows GATE's output document annotated with the entities "*Location*" and "*Person*" when GATE is embedded in a java application and Fig 4.3 shows the GATE output document annotated with entities "*Bird*" and "*Location*" saved as a XML file.

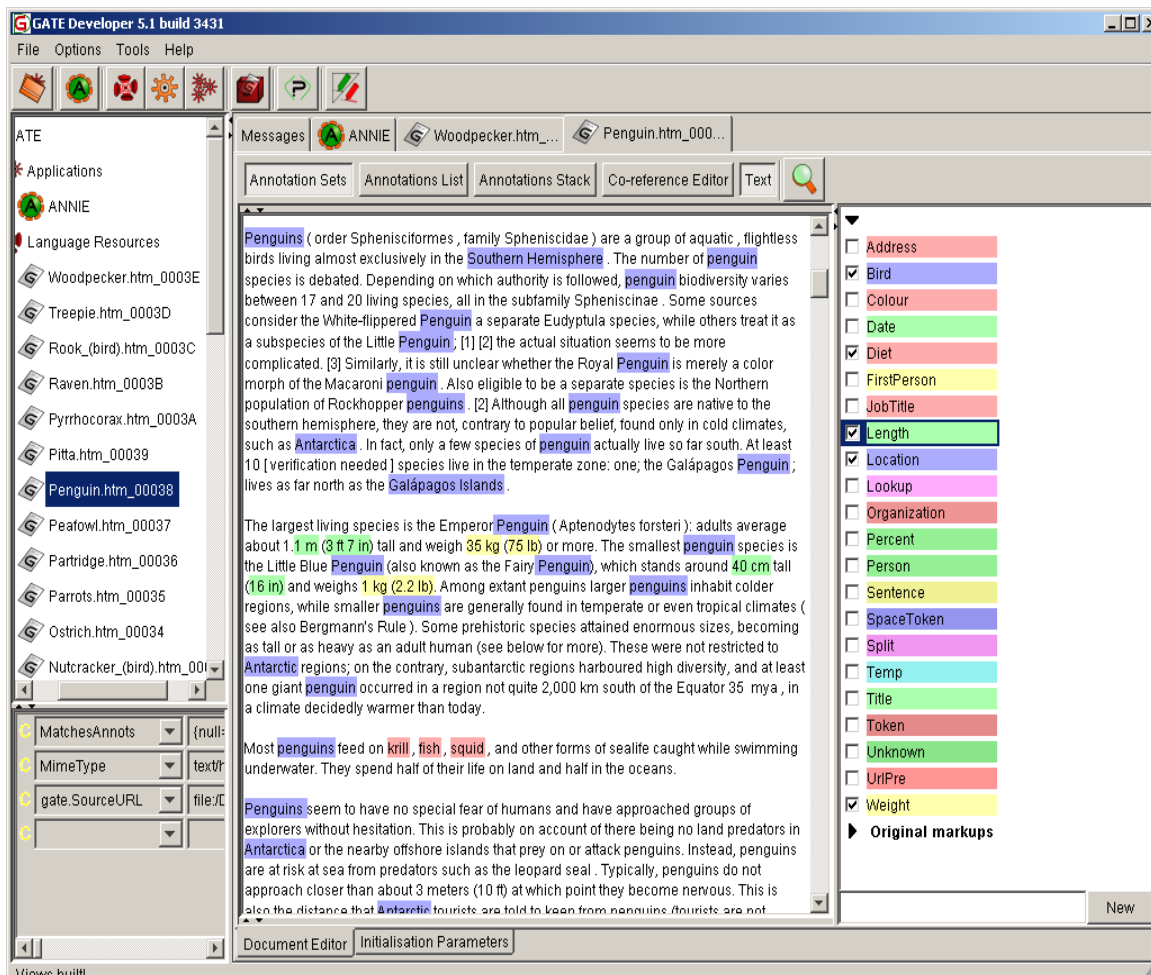


Fig 4.1 GATE's User Interface with an annotated document

An extreme example of this was the [Emu War](#) in [Western Australia](#) in 1932, when Emus that flocked to Campion during a hot summer scared the town's inhabitants and an unsuccessful attempt to drive them off was mounted. In [John Gould's](#) *Handbook to the Birds of [Australia](#)*, first published in 1865, he laments the loss of the Emu from [Tasmania](#), where it had become rare and has since become extinct; he notes that Emus were no longer common in the vicinity of [Sydney](#) and proposes that the species be given protected status.^[7] Wild Emus are formally protected in [Australia](#) under the [Environment Protection and Biodiversity Conservation Act 1999](#). The [IUCN](#) rates their status as [Least Concern](#).^[1] Their occurrence range is between 1,000,000–10,000,000 km² (390,000–3,900,000 sq mi), and a 1992 population estimate was between 630,000 and 730,000.^[29]

Fig 4.2 Part of a text from GATE's output annotated with the entity *location*.



Fig 4.3 GATE output saved as a XML file.

4.3 Natural Language Parsing Towards Relation Exaction

Verb is the powerful lexical term which binds two adjacent syntactic categories and a relation can be defined as a predicate expression of two nouns i.e. subject and object wrapped in syntactic categories as follows.

Verb(Subject, Object) or Verb_Prep(Subject, Object)

Verb_prep is the form of the verb combined with a preposition. Therefore the identification of the main verb in a sentence is promising initiative in defining a relation between two entities. But extraction of verb constituent from natural language text which relates two entities demands heavy linguistic processing. For the purpose of relation extraction by verb predicate, documents should be parsed in to identified sentence structures and then the main verb predicates can be derived from the sentence structure.

For an example the sentence “*Jackdaws are found in Europe, Iran, north-west India and Siberia where they inhabit wooded steppers, woodland, cultivated land pasture, coastal cliffs and villages*” can be mapped to the above predicate format as follows after the sentence is parsed and tagged for syntactic constituents and concepts.

located_in(Jackdaw, Europe), located_in (Jackdaw, Iran),
located_in(Jackdaw, north-west India), located_in(Jackdaw, Siberia)
Inhabit(Jackdaw, woodland), Inhabit(Jackdaw, wooded steppers)
Inhabit(Jackdaw, cultivated land pastures), Inhabit(Jackdaw, coastal cliffs),
Inhabit(Jackdaw, villages),

Semantic ambiguity (i.e. one word may have multiple meanings (polysemy) and several words can have the same meaning (synonymy)) is one of the difficulties that comes in natural language processing. Therefore semantic disambiguation should be addressed in order to extract accurate information from the natural language text. It is not possible to assign either one verb constituent or pre-defined set of verbs for a relation. Semantic ambiguity is dealt by keeping a set of equivalent verbs for a relation and updating the set whenever evidence to add a new verb to the set is found. For an example in the sentences (1) and (2) shown in below Section 4.3.1, verbs “*found in*” and “*are native*” lead the way to the relation “*located_in*” . Therefore “*are native*” and “*found in*” can be considered as equivalent terms (not synonyms) for “*located_in*” under background information.

Normally sentences do not always exist in a simple structure in natural language. There can be more than one verb constituent in a sentence from which the verb that binds two entities should be identified. Sometimes the main verb constituent of the sentence does not account for the relation between the two identified entities depending on the complexity of sentences. A relationship between entities is denied by negative sentences and those sentences can be identified by a negative word appearing with a verb constituent (e.g. *not*, *never* etc). But some verbs can be considered as negative verbs for a relation (e.g. *extinct* for *located_in*). Therefore a sentence should be carefully analyzed in order to find whether it actually gives a relation or not.

4.3.1 Reducing Stanford Dependencies for Relation Extraction

Stanford parser is used on GATE output which is annotated with the entities, to identify syntactic constituents of a sentence and to derive dependencies among them. In parsing, annotations are removed from the sentences and lists of entity types are appended with the annotated items for the purpose of annotating terms again when enclosed in dependency clauses.

For an example for the following two sentences of different syntactic structures and similar semantics the Stanford parser gives syntactic tagging and the dependencies as given below. Graphical representation of sentence 1 is given in the fig. 4.4.

Sentence 1 - “Humming Birds can be found in Cuba including Isle of Youth” ----- (1)

Tagging

Humming/NNP, Birds/NNP, can/MD, be/VB, found/VBN, in/IN, Cuba/NNP, including/VBG, Isle/NNP, of/IN, Youth/NN

Universal dependencies

compound(Birds-2, Humming-1)
 nsubjpass(found-5, Birds-2)
 aux(found-5, can-3)
 auxpass(found-5, be-4)
 root(ROOT-0, found-5)
 case(Cuba-7, in-6)

```

nmod(found-5, Cuba-7)
case(Isle-9, including-8)
nmod(Cuba-7, Isle-9)
case(Youth-11, of-10)
nmod(Isle-9, Youth-11)

```

Universal dependencies, enhanced

```

compound(Birds-2, Humming-1)
nsubjpass(found-5, Birds-2)
aux(found-5, can-3)
auxpass(found-5, be-4)
root(ROOT-0, found-5)
case(Cuba-7, in-6)
nmod:in(found-5, Cuba-7)
case(Isle-9, including-8)
nmod:including(Cuba-7, Isle-9)
case(Youth-11, of-10)
nmod:of(Isle-9, Youth-11)

```

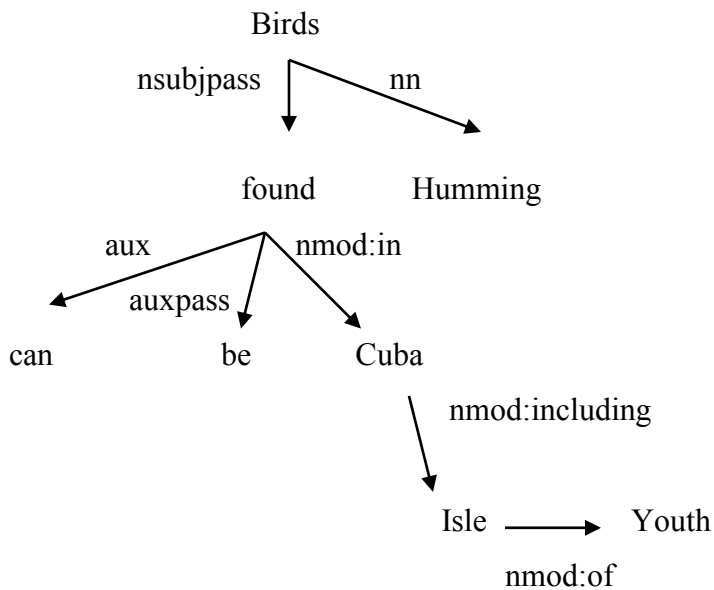


Fig 4.4 Graphical representation of Stanford dependencies for the above sentence.

Sentence 2 – “Ostriches are native to savannas and the Sahel of Africa, both north and south of the equatorial forest zone.” -----(2)

Tagging

Ostriches/NNS, are/VBP, native/NN, to/TO, savannas/NNS, and/CC, the/DT,
Sahel/NNP, of/IN, Africa/NNP,, /,, both/DT, north/RB, and/CC, south/RB, of/IN,
the/DT, equatorial/JJ, forest/NN, zone/NN

Universal dependencies

nsubj(native-3, Ostriches-1)
cop(native-3, are-2)
root(ROOT-0, native-3)
case(savannas-5, to-4)
nmod(native-3, savannas-5)
cc(savannas-5, and-6)
det(Sahel-8, the-7)
conj(savannas-5, Sahel-8)
case(Africa-10, of-9)
nmod(Sahel-8, Africa-10)
cc:preconj(north-13, both-12)
advmod(native-3, north-13)
cc(north-13, and-14)
conj(north-13, south-15)
case(zone-20, of-16)
det(zone-20, the-17)
amod(zone-20, equatorial-18)
compound(zone-20, forest-19)
nmod(native-3, zone-20)

Universal dependencies, enhanced

nsubj(native-3, **Ostriches**-1)
cop(native-3, are-2)
root(ROOT-0, native-3)
case(savannas-5, to-4)
nmod:to(native-3, savannas-5)
cc(savannas-5, and-6)
det(Sahel-8, the-7)
nmod:to(native-3, Sahel-8)
conj:and(savannas-5, Sahel-8)
case(Africa-10, of-9)
nmod:of(Sahel-8, **Africa**-10)
cc:preconj(north-13, both-12)
advmod(native-3, north-13)

cc(north-13, and-14)
advmod(native-3, south-15)
conj:and(north-13, south-15)
case(zone-20, of-16)
det(zone-20, the-17)
amod(zone-20, equatorial-18)
compound(zone-20, forest-19)
nmod:of(native-3, zone-20)

Highlighted terms in both dependencies indicate the already identified entities by the use of GATE and all the terms are syntactically tagged by the parser.

From both sentence structures 1 and 2 the relation extracted should be in the form

located_in(Humming Bird, Cuba)

located_in(Humming Bird, Isle of Man)

located_in(Ostriches, Africa)

When generating rules for relation extraction only the dependencies which involve relevant entities identified by GATE and verb constituents in the sentence have to be dealt. Therefore the dependencies are preprocessed to filter the relevant atomic formulas which can contribute to the rule formation. Relevant atoms contain at least one entity instance. The tense, the subject number or the voice of the sentence are not taken into account. All the nouns and verbs tagged originally by the parser are given category NN (stands for a noun) and VB (stands for verb in the base form) respectively. Therefore all the nouns and verbs annotated as NNS, NNP, VBS, VBP and VBG [Appendix E] etc. are simplified to either NN or VB. The syntactic categories and entity types are needed to be assigned to each term enclosed by dependency clauses in generating extraction rules. When sentences grow in complexity and length the dependencies tend to be complicated and vast. Therefore by considering scope of the task some measures are taken in order to reduce the complexity of the dependencies of a sentence.

- The atom “*nsubj*” is the nominal subject which is the subject noun phrase and “*nsubjpass*” is the passive nominal subject in Stanford dependencies.[see Appendix F]. Therefore the atom “*nsubjpass*” is conveniently replaced by “*nsubj*”. The main verb of the sentence is contained in “*nsubj*” or “*nsubjpass*” clauses.
- The atom “*nmod:including*” which enclose two instances of same entity is replaced by “*conj:and*” as both atoms give similar dependencies. Two adjacent noun

constituents or adjective and noun constituent in atoms “*nn*” and “*nmod:of*” are considered as one term on similar grounds.

- Adjective and noun contained in Adjectival modifier “*amod*” which is any adjectival phrase that serves to modify the meaning of the NP, are considered as one term if the noun is a domain entity. Then the dependency “*amod*” is omitted from the background of the dependencies of the training sentences for most of the relations. But for some relations “*amod*” is required to capture a relationship. When “*amod*” encloses an adjective and a domain entity, it is the most significant clause which describes the characteristics of the entity. Therefore “*amod*” is not omitted from the dependencies of the test corpus. If the adjective in “*amod*” is an entity it is not omitted from the dependencies of the sentences in any of the corpus. When joining the terms in the clauses “*nn*”, “*nmod:of*” and “*amod*”, the clause “*nn*” is given the priority, then “*nmod:of*” and last “*amod*”

- Since only the relation verb which binds two entities are considered, the clauses “*advcl*” and “*advmod*” which modifies a verb or the meaning of a verb can be ignored. But in the same time these clauses can be used as in the case with *amod* for the relations which describe an action; for examples, **has_characteristic(Ostrich, run_fast)** or **run(Ostrich, fast)**.

- Auxiliary verbs “*aux*” and “*auxpass*” which are non-main verbs of the clause such as “be”, “have” etc. are combined with the relevant verb.

- Atoms that represent adjectives, adverbs and determinants are ignored because there is no significant impact on relations by them.

- If a verb constituent is missing in “*nsubj*”, typed dependencies are searched through to find the verb associated with the noun constituent in “*nsubj*”.

- The atoms “*det*” and “*predet*” are ignored as it indicates the determinants.

- A dependency labeled as “*dep*” is ignored because *dep* is a very general dependency and used when the system is unable to determine a more precise dependency relation between two words.

- The *case* marker dependency which indicates prepositions is also ignored because *nmod* totally takes care of prepositions.

- The clause “*cop*” which encloses a copular verb contributes a great deal for identification of taxonomic relations. However, in non- taxonomic relations “*cop*” can be omitted unless it contains an entity. Furthermore, in non-taxonomies, when the clause “*nsubj*” does not enclose a verb constituent, copular verb is combined with the relevant noun in the clause in order to make the relation verb. Since the clause “*cop*” is needed in identifying taxonomic relations “*cop*” is not removed from the dependencies of the sentences in extracting relations.

When a verb is joined with a noun the whole term is assigned the syntactic category VB.

For example the reduced typed dependencies of the above mentioned sentences are shown below with the graphical representation of reduced dependencies of sentence (1) in fig.4.5. Overall dependency reduction process is given in fig.4.6.

Sentence (1)

nsubj(can_be_found<VB>-5, **Humming_Birds**<NN><Bird>-2)

nmod:in(can_be_found<VB>-5, **Cuba**<NN><Location>-7)

conj_and(**Cuba**<NN><Location>-7, **Isle_of_Youth**<NN><Location>-9)

Sentence (2)

nsubj(are_native<VB>-3, **Ostriches**<NN><Bird>-1)

Conj:and(are_native<VB>-3, **Sahel_of_Africa**<NN><Location>-8)

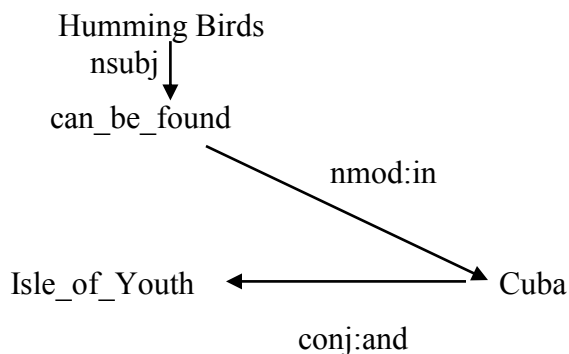


Fig 4.5 Graphical representation of reduced dependencies of Sentence (1)

Sentences annotated with only one entity are treated for deriving taxonomical relations. Then the super-class of the annotated entity should be identified. The following sentence No.3 falls into that category. “*Ostrich is a flightless bird.*” ----- (3)

The Sentence No. 3 which leads to the extraction of relation “ *is_a(Ostrich, flightless bird)*” with the following reduced typed dependencies.

nsubj(bird<NN>-5, **Ostrich**<NN><**Bird
 cop(flightless_bird<NN>-5, is<VB>-2)**

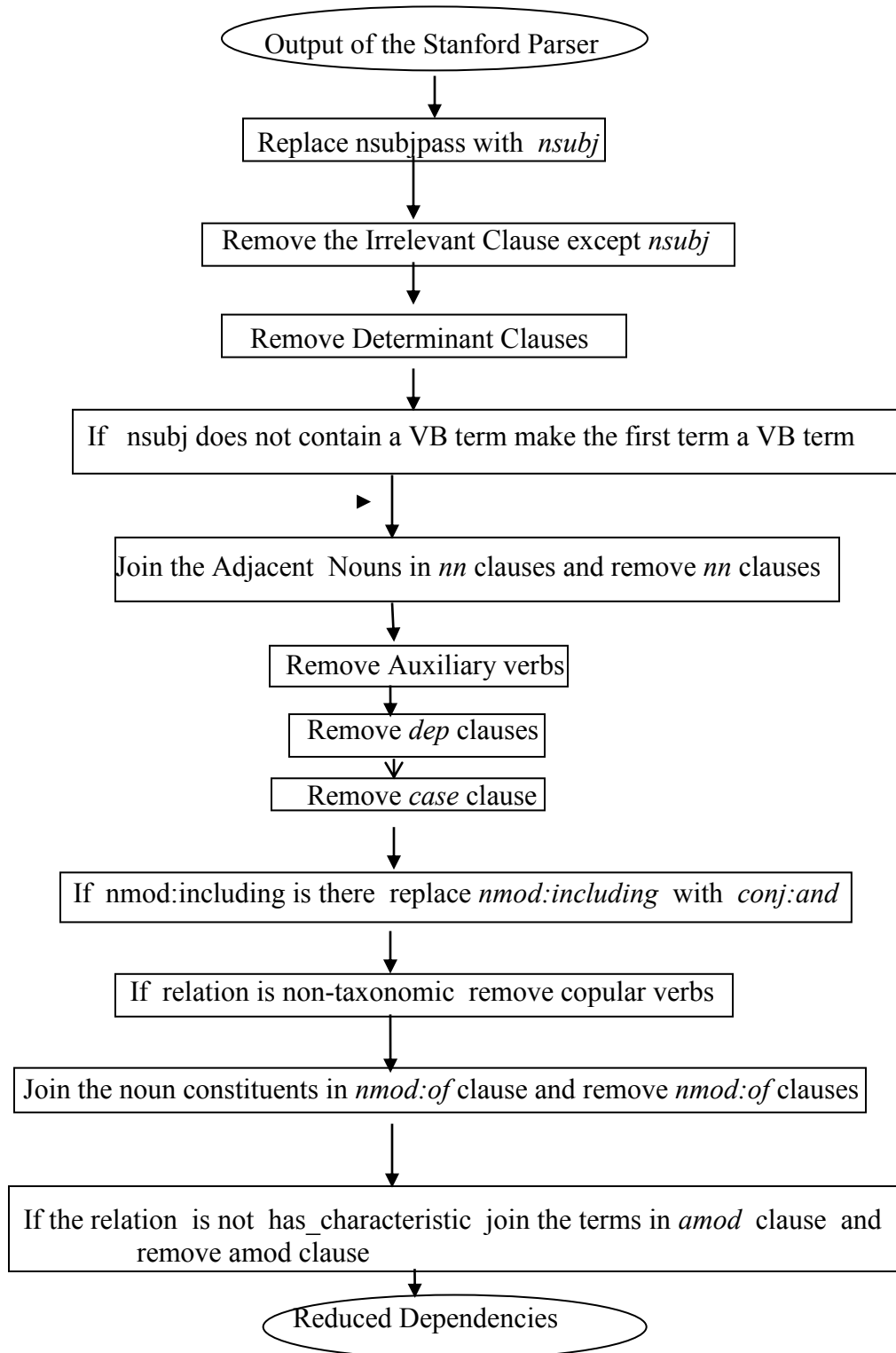


Fig 4.6 The Process of generating Reduced Dependencies from Stanford dependencies of an English Sentence

4.3.2 Identifying Conjunctions and Relative Clauses

Extracting information accurately, from the compound sentences and sentences which consist of relative clauses or conjunctions should be addressed further, paying attention to connecting words used in each of those categories. In compound sentences two independent clauses are connected by compound words. Therefore sentence constituents connected by compound word can be processed separately and all the constituents in the sentence can be considered as true statements in the normal way as simple sentences. Since relative clauses describe nouns and are directly addressed by the dependencies, further processing is not needed in generating relation-extraction-rules.

But with conjunctions an extra effort is needed to retain the accuracy of the information given by the sentence because truth value of the sentence depends on some conjunctions. Some conjunctions are strictly conditional and have a direct effect on the truth value of the information given by the sentence depending on the truth value of the constituent coming with the conjunction. Sentences can be made more informative with some conjunctions and truth value of the sentence does not depend on the truth value of the conjunctive part. Therefore unconditional conjunctions can be handled in the same way as compound sentences.

Commonly used conditional and unconditional conjunctions are given in table 4.1

Conditional Conjunctions	Unconditional Conjunctions
If When As long as Before After Provided Until Or	While Though Although Whether And

Table 4.1 Commonly used conjunctions

Conditions come with the conjunctions can be considered as restriction for the ontological entities and relations to be true. Information wrapped in the sentence fragment with the conditional conjunction can be captured separately and presented as a condition/restriction for information given by the other fragment of the sentence. For an example the sentence *When being pursued by a predator, ostriches have been known to reach speeds in excess of 70 km/h* gives the information that ostriches can have a

minimum speed 70 km/h under the condition “pursued by a predator”. Then the relation instance *has_speed(Ostrich, 70_km/h)* will be true if the condition *pursued_by(Ostrich, predator)* holds. Fig. 4.7 shows the way the sentences are categorized with respect to information extraction.

In extracting the condition the dependencies are searched to find the verb constituent in the conjunctive part. Then the verb is taken as the relation predicate and it is attributed with subjective noun or an entity where applicable and the closest noun to the conjunctive verb.

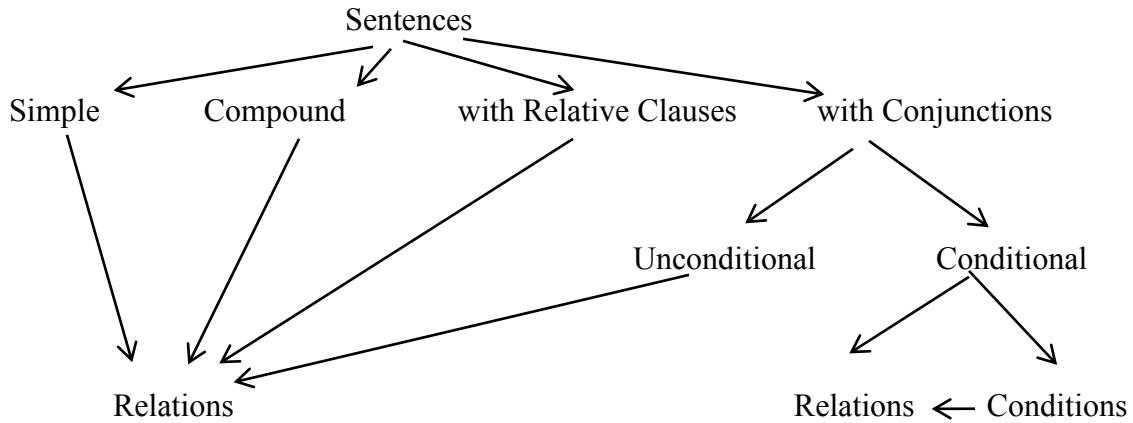


Fig 4.7 Main Sentence Categories with respect to Information Extraction

4.4 Inductive Logic Programming for Generation of Relation-extraction-Rules

.Inductive logic programming (ILP) uses mathematical logic in computer programming as a uniform representation for examples, background knowledge and hypotheses [34]. Given an encoding of background information and set of positively and negatively labeled examples represented in the form of logical database of facts, ILP derives hypotheses which entail all positive and none of the negative examples. Many applications benefit from the relational descriptions generated by the ILP systems and the applicability is enhanced by the accommodation of background knowledge [34] .

Relation extraction methodology that have been developed involves number of complex processes, learning rules from positive training data, modifying learned rules by applying them on negative data, verification of the modified rules, application of learned rules on

test data for successful relation extraction, updating the positive and negative verb sets for the relation, identifying new relations existing between known entities etc. Stanford parser is used on GATE output which is annotated with the entities, to identify syntactic constituents of a sentence and to derive dependencies among them. The training data set to be used by the methodology developed for generating relation-extraction-rules contains reduced dependencies along with syntactic tags of relevant sentences. Then a technique which involves ILP is used to induce rules for relation extraction, searching through the dependencies of natural language sentences given in the training set. The dependencies and syntactic tags provide background knowledge to learn rules for relation extraction. While the task of the rule learning is to learn rules from the training data, the main task of the relation extraction is to find the relation instances by applying the learned rules.

The rule learning process uses the output of the Stanford parser to learn rules to extract relation instances for a known relation such as *located_in*, *part_of*, *feed_on* etc., and some of which are domain specific relations. It employs inductive logic programming (ILP) technique [34] in the learning algorithm to derive the set of rules from the dependencies based on the text annotated with the entities. The Inductive Logic Programming technique that have being adopted for the rule learning task is explained in detail in the Section 4.4.4 and 4.4.6. If necessary the learned rule set is updated whenever the training data set is appended with new information by the user. The set of equivalent verbs for a relation is also updated with new found verbs and new relations existing between the entities are identified during the relation extraction process that is explained in detail in 4.4.2. Extraction rules are generated for both taxonomic and non-taxonomic relations after the sentences are identified for both relation categories.

Since the work described in this theses started with the Stanford typed dependencies it is preferred to use the names of typed dependency clauses in the rules. There is only a small difference between terminology of universal dependencies and typed dependencies. For an example all the prepositional clauses are labeled with *prep* instead of *nmod*, i.e. *nmod:in* is replaced with *prep_in*.

4.4.1 Taxonomic Relations

Sentences annotated with one or more entity types are preprocessed to see whether it contains ingredients for taxonomic relations. The existence of a domain name or an entity class name in a sentence is considered as required features for taxonomic relations.

For an example the sentence “*Ostrich is a flightless bird*” gives the taxonomic relation *is_a(Ostrich, flightless bird)* and the existence of the domain name bird in the sentence makes it an eligible sentence for the consideration of a taxonomic relation. The extended version of the above sentence annotated with bird and location “***Ostrich*** is a flightless bird found in ***Antarctica***” gives two relations

located_in(Ostrich, Antarctica) and

is_a(Ostrich, flightless bird)

Stanford dependencies provides clauses specific to Hearst patterns [25] which are used in identifying taxonomic relations. Those clauses are also then used to generate extraction rules for taxonomic relations.

4.4.2 Non Taxonomic Relations

Sentences annotated mainly with two entities are treated to extract non taxonomic relations. Two entities can be two nouns connected by a verb constituent which denotes the relationship between the two entities. The challenging task is identifying the correct verb constituent which binds the two entities. When the sentence is complicated and long the relation verb cannot be readily found. Sometimes it has to be filtered from other verbs in the sentence.

From the available training examples, a set of positive verbs and a set of negative verbs for a relation can be created. In identifying the correct relation verb there are number of issues that should be taken into consideration. The main verb constituent of a sentence is normally wrapped in atomic formula *nsubj*. But when the sentence gets more complicated and contains more than one verb, identifying the relation verb of the annotated entities demands additional facts for correct identification. In such situations we can make use of the positions of the words in a sentence given by the parser. The minimum difference of the distances to both entities from each verb is considered as a measure in selecting the relation verb.

For an example, the sentence “*Numerous **ducks** have managed to establish themselves on oceanic islands such as **Hawaii** and **New Zealand***” contains two verbs “*managed*” and “*establish*”. Reduced dependencies obtained as explained in the Section 4.3 from the dependencies of the sentence generated by the Stanford parser are given below.

```
nsubj(managed-4, Numerous ducks-2)
xsubj(establish-6, Numerous ducks-2)
xcomp(managed-4, establish-6)
dobj(establish-6, themselves-7)
prep_on(establish-6, oceanic_islands-10)
prep_such_as(oceanic_islands-10, Hawaii-13)
prep_such_as(oceanic_islands-10, New Zealand-17)
conj_and(Hawaii-13, New Zealand-17)
```

The above sentence is a good example for the relation *located_in*. But the main verb constituent is clearly not an equivalent verb for the said relation. The verb “*establish*” can be considered as a suitable equivalent for the relation *located_in*. The difference of the distances to two entities *ducks* and *Hawaii* from the verb “*managed*” is 7 whereas the difference of the distances from the verb “*establish*” is 3. Therefore the verb “*establish*” with the minimum difference is added to the set of positive verbs for the relation.

When there is a situation of conflict where two verbs give the same difference value none of the verbs is considered as a positive verb and the situation is left to be solved by human involvement.

When there are two “*nsubj*” atoms in the dependencies and two entities are divided in between two sections of “*nsubj*” the verb in the second section is taken as the positive verb.

When *nsubj* contains a non-verb attribute with an entity, non-verb attribute is added to the set with a copular verb. For an example in the sentence “*Ostriches are native to Africa*” the word “*native*” is tagged as a noun and typed dependencies of sentence contains

```
nsubj(native-3, Ostriches-1)
cop(native-3, are-2)
```

Then the verb is identified as “*are native*” and both “*are native*” and “*is native*” are added to the positive verb set.

Relations which require identification of noun or adjective constituent can also be established. The relation “*has_characteristic()*” identifies special characteristics of an

object or a person. This relation is used in the example domain *Bird* which is experimented in the research work present in the thesis, to find the features of body parts of a bird. Therefore rules are needed to extract adjectives which describe body parts of a bird. Then the sentences annotated with entity “*Part*” are considered for this relation.

For example from the sentence “*Choughs have long broad wings and perform spectacular aerobatics*”

the relation *has_characteristic*(*Chough*, *long_broad_Wing*) can be extracted.

The reduced dependency of the sentence is given below.

nsubj(have-2, Choughs-1)

dobj(have-2, long_broad_wings-5)

When there is more than one adjective to describe an object all the adjectives can be combined according to the order given by the position number of adjective term.

4.4.3 Extraction of Ontological Relations

The rules generated are applied to extract relations in a given corpus of a particular domain. The Stanford dependencies of sentences of the known entities in the document are searched to find the compatibility of an extraction rule with the dependencies. Entity instances in sentences covered by the rules of a particular relation are extracted as the attribute values of that relation. When a sentence cannot be covered by extraction rules, the positive and negative verb sets for relations are searched in order to find out whether the main verb constituent is equivalent to any of the verbs in the two sets. Sentences of entities not extracted as a relation by existing relation-extraction-rules, can be processed in order to find whether the entities form a negative relation or a new relation. An ambiguous sentence with respect to extraction rules can be categorized into one of the following situations.

- (i) Verb unknown but extraction rules cover the dependencies
- (ii) Verb known but extraction rules cannot cover the dependencies.
- (iii) Verb unknown and extraction rules cannot cover the dependencies.

Sentences that fall into group (i) are considered as positive candidates for a respective relation. Then, most appropriate verb constituent in the dependencies of the sentence is

considered as the relation verb and added to the set of positive verbs for the relation. Sentences in category (ii) give a different structure and the main verb in the sentence confirms that the sentence contains ingredients for the relation. Then that sentence is used to form a rule to cover newly found sentence structure.

Sentences in category (iii) are assumed to be formed a completely new relation and they are used to formulate the new relation. The relation is labeled by the main verb of the sentence; i.e. the verb constituent contained in the atomic formula “*nsubj*” when there is only one verb in the dependencies. For an example the sentence “Green pheasant is the national bird of Japan” falls into category (iii) on the application of extraction rules for relation *located_in* and it gives a new relation *is_national_bird*.

In adding the equivalent verbs to both positive and negative sets there is a possibility of adding a different grammatical form of the same verb because the verb is simply taken as it appears in the sentence. For an example the verb “*find*” can be added to an equivalent verb set when the verb “*is_found*” is already in the set.

When applying rules for taxonomic relations, correct identification of the presence of a super class in the sentence is very important to avoid extraction of incorrect relations. A super class is identified by a domain name or an established entity name. For an e.g. in the sentence “*Ostrich is a flightless bird*” the word “*bird*” is a domain name as well as an entity name

Fig. 4.9 shows the intakes to both processes rule learning and relation extraction with the outcomes of the tasks.

4.4.4 Adaptation of Attribute Value learning for Ontological Relation

ILP techniques are used on the output of the Stanford parser to generate rules for relation extraction. Since Stanford parser provides many atomic formulas or atoms (i.e. predicate expression with two tuples) in the form of dependencies as well as syntactic tagging, the output of the Stanford parser is a good candidate for inductive logic programming. A set of positive and negative training examples for a relation can be made available along with syntactic constituents (syntactic tags) of the sentence from which the relation is extracted. For example the sentence *Humming Birds can be found in Cuba including Isle of Youth* gives a positive instance for the relation *located_in* resulting

located_in(Humming Bird, Cuba) and

located_in(Humming Bird, Isle of Youth)

The sentence *Cranes live on all continents except Antarctica and South America* is an evidence for negative relation instances and the extraction can be represented as

$\neg \text{located_in}(\text{Cranes}, \text{Antarctica})$ and

$\neg \text{located_in}(\text{Cranes}, \text{South America})$.

Therefore the set of positive training data E^+ , the set of negative training data E^- and the background information B can be defined for ILP as follows

$E^+ = \{\text{Positive relation instance pairs for the relation}\}$

$E^- = \{\text{Negative relation instance pairs for the relation}\}$

$B = \text{Reduced Stanford parser output, Syntactic tags}$

Fig. 4.8 (a), Fig. 4.8 (b). and Fig. 4.8 (c) show how an initially generated single rule, modified rule and the final set of refined rules cover the training data.

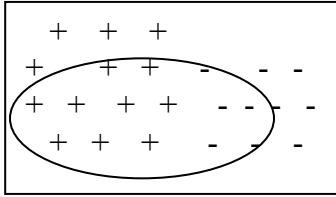


Fig. 4.8 (a) Initial Rule

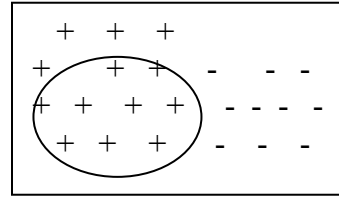


Fig. 4.8 (b) Modified Rule

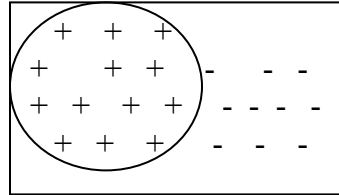


Fig. 4.8 (c) The Final Set of Rules.

ILP algorithm is adapted from the attribute value learning system LINUS[34] with an approach similar to NEWGEM propositional learner[34], to induce rules from the available atoms given by Stanford parser in order to cover all the positive training data. LINUS induces hypothesis in the form of constrained deductive hierarchical database (DHDB) clauses. The main idea in LINUS is to transform the problem of learning relational DHDB descriptions into a propositional learning task and incorporate a suitable propositional learner to induce rules for the required task. Since the background of our problem contains atomic formulas it is already in the form of a propositional learning

task. Then it is only required to incorporate a suitable propositional learner in the LINUS system in order to learn rules from the training data and the background knowledge.

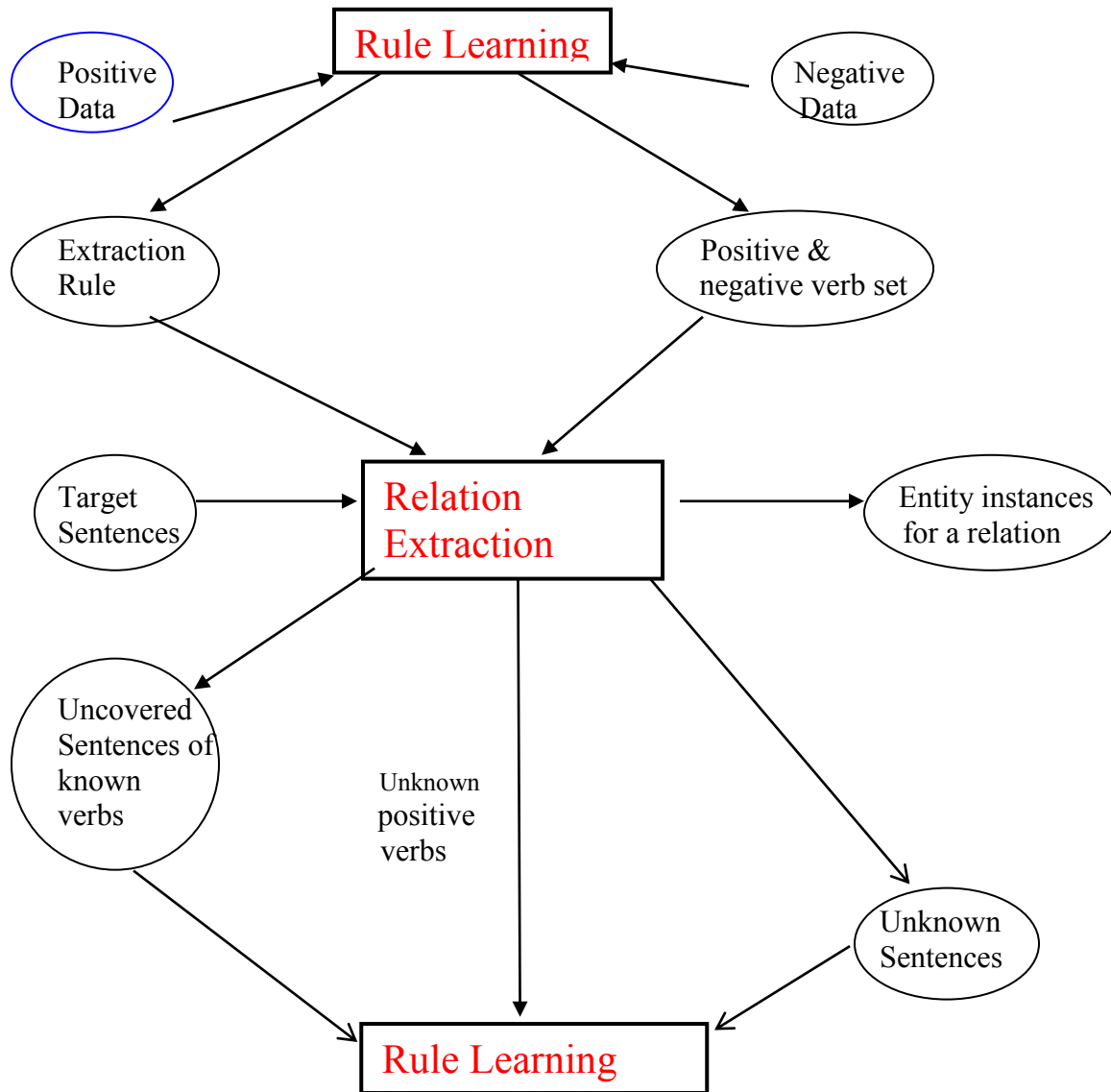


Fig 4.9 Input and output of the two processes Rule Learning and Relation Extraction

LINUS Algorithm that is used to generate relation extraction rules

Preprocess the training set to establish the sets E^+ and E^- of positive and negative examples.

Process both positive and negative training sets to

- create a list of atoms ordered according to the number of occurrences from E^+
- create a list of atoms ordered according to the number of occurrences from E^-
- create a set of positive verbs for the relation.
- create a set of negative verbs for the relation.

Use an attribute value learner to induce extraction rules.

Transform rules into DHDB clauses.

4.4.5 Processing the training set.

The availability of negative examples in the training set depends on the domain document corpus. In some domain documents, negative relations are readily available. When negative relations are not available in document sentences, any other relation which binds two domain entities within a sentence and is not annotated for the relation concerned is considered as a negative relation. Sentences that contain one entity are treated for taxonomical relations. But there isn't a guarantee that documents in any domain will give sentences to generate negative examples from the selected sample of training documents. In such cases negative examples should be constructed by making some positive examples negative or using negative verbs.

Reduced dependencies of all the positive sentences are first processed to count the number of times an atom appears in dependencies. All the atoms found are placed in a set according to the order of occurrence (i.e. the number of times an atom occurs in typed dependencies). Same treatment is applied to negative sentences and a set of ordered atoms is created for negative sentences. Then the set of atoms relevant only to the negative sentences that is required in modifying the original rules can be filtered out by the difference of two sets. Establishing two ordered sets for both positive and negative atoms minimizes the efforts in refining the extraction rules.

4.4.6 Processing positive and negative training data using the attribute value learner

A relation is represented by the consequence of the rule as an attribute value tuple and the body of the rule is by the antecedent of the rule. The body of the rule contains one or more conditions for the relation to be fulfilled when applied on reduced Stanford dependencies. Since dependencies are already in the propositional form they can be directly applied to ILP. Rules are specialized with respect to negative examples by adding atoms from dependencies of negative data if the rules can extract any of the negative relations. In deriving the preliminary set of rules the atom relevant to subject noun, *nsubj* is taken out from the set of available atoms as it is present in almost all the sentences. It comes as the first condition in every rule. Rules are initially formed with two atoms; *nsubj* and another atomic formula from the dependencies. First attribute in *nsubj* is identified mostly as a verb, but rarely a noun or an adjective and the second attribute can be one of the annotated entities. Then a rule is in the following format.

$$nsubj(Verb/Noun/Adjective, Entity1/Entity2) \wedge atom(...., \Rightarrow) \quad Relation(Entity1, Entity2)$$

But there are situations where *nsubj* does not contain a relation verb or a domain specific entity. Sentences in such situations are handled separately to form the rule with any two clauses which involves entities and the relation verb.

Separate rules are constructed at the disjunction between conditions. Rule body allows only internal disjunction (i.e. disjunction between attribute values) that appears in a rule. Table 4.2 shows combination of atomic formulas of three positive example and one negative example from the relation *located_in(bird, location)* where *bird* and *location* are entity classes. Definitions of the atomic formulas are given in the Appendix F and a sample of training examples with the reduced dependencies are given in Appendix G. The variables *x* and *y* are used to represent entities. In the absence of entities they represent syntactic tags. For relation examples a relation is considered as a class for ILP method.

The representation of dependencies is already in the propositional form and can easily be converted into a way suitable for attribute value learning. The attributes and the values from the propositional form for the target relation *located_in(bird, location)* are shown in

Table 4.3. It shows the instances of entity classes, *bird* and *location* for the three positive examples and one negative example along with the dependencies relevant to each example.

Class	nsubj(x,y)		prep_to(x,y)		conj_and(x,y)		prep_in(x,y)		prep_except(x,y)	
	X	y	X	y	X	y	x	y	x	y
located_in(bird, location)	Verb	bird	verb	location	verb	location				
located_in(bird,,location)	verb	bird					verb location	location location		
located_in(bird,location)	verb	bird			location	location	verb verb	location location		
¬located_in(bird,,location)	verb	bird					verb verb	noun location	verb	location

Table 4.2 Examples of combination of atomic formulas for the relation “*located_in*”

Entities can be considered as attributes, and instances of entities give the values. The variables *x*, *y* denote the entity class instances while the variable *z* indicates the syntactic category of another term which associates with entity instances. The presence of entities in dependency clauses is the major factor in identifying a relation and the lexical represented by *z* does not play a significant role here.

The rule generation in the algorithm is not initiated by a seed (i.e. a positive example) as in NEWGEM. Instead it collects elements (i.e. atomic formulas from dependencies) for the rule formation from the set of ordered atoms for positive examples and places them in a list. A heuristic approach is used to create the list in finding the atomic formula to combine with *nsubj* to form rules. Most occurring atoms are given the priority to join with *nsubj* to form a rule and the most generalized rule is formed first. The list of atoms is created from the set sorted according to the number of times that an atom occurs. In NEWGEM algorithm a set of alternative rule bodies which maximize the number of covered positive examples is defined as a beam. In algorithm developed here, the beam is the list of atoms and not a set of rule bodies. Since the objective is to form the rules with best combination of atoms, rule bodies are formed, taking atoms from the beam to combine with *nsubj*. Then the formed rules are specialized with respective to negative training data. Therefore in finding the best body, the best atom from the beam is taken to form the rule body. Table 4.3 shows the existence of atomic formulas in the Stanford

dependencies for three positive and one negative instance for the relation *located_in*, taken from the training examples given in the Appendix G.

C l a s s	Variables			Propositional Features						
	Bird X	location y	Other z	nsubj (z,x)	prep_to (z,y)	conj_and (z,y)	conj_and (y,y)	prep_in (z,y)	prep_in (y,y)	prep_except (z,y)
+	Ostri ch	Savannas, Sahel_of_Af rica	Verb	True	True	true	False	False	False	False
+	Hum min g_Bi rd	Cuba, Isle_of_Yout h	Verb	True	False	False	False	True True	False	False
+	Parr ots	America, Australasia	Verb	True	False	False	True	True True	False	False
-	Poot oos	Chile	Verb Noun	true	False	False	False	True	false	True

Table 4.3 Some positive and a negative entity instances for the relation “located_in”

Appendix G includes the reduced dependencies for some positive examples and negative examples that were used to form the initial set of rules.

From the positive examples shown above in the table 4.2, the initial set of rules can be formed as follows.

$nsubj(verb, Humming\ bird) \wedge \neg prep_in(verb, Cuba) \longrightarrow located_in(Humming\ Bird, Cuba)$

$nsubj(verb, Humming\ Bird) \wedge prep_in(verb, Isle_of_Man) \longrightarrow located_in(Humming\ Bird, Isle_of_Man)$

$nsubj(verb, Parrots) \wedge prep_in(verb, America) \wedge conj_and(America, Australasia) \longrightarrow located_in(Parrot, America)$

$nsubj(verb, Parrot) \wedge prep_in(verb, Australasia) \wedge conj_and(America, Australasia) \longrightarrow located_in(Parrot, Australasia)$

$nsubj(verb, Ostrich) \wedge prep_to(verb, Savannas) \longrightarrow located_in(Ostrich, Savannas)$

$nsubj(verb, Ostrich) \wedge conj_and(verb, Sahel_of_Africa) \longrightarrow located_in(Ostrich, Sahel_of_Africa)$

The number of rules can be reduced by generalization. In generalizing, some rules become redundant and some rules can be joined together by internal disjunction.

From Table 4.2 and Table 4.3 the Table 4.4 that is the base for generalization of the rules can be generated.

Class	Variable	Propositional Features						
	Syntactic Constituent Z	nsubj (z,bird)	prep_to (z,location)	conj_and (z,location)	conj_and (location,location)	prep_in (z,location)	prep_in (location,location)	prep_except (z,location)
+	Verb	True	True	True	False	False	False	False
+	Verb	True	False	False	False	True	True	False
+	Verb	True	False	False	True	True	false	false
-	Verb Noun	True	false	True	place	True True	false	true

Table 4.4 Generalization of the data shown in table 4.3

The following shows the generalized form of the above mentioned rules

R1: $nsubj(z,Bird) \wedge prep_in(z,Location) \longrightarrow located_in(Bird,Location)$

R2: $nsubj(z,Bird) \wedge prep_to(z,Location) \longrightarrow located_in(Bird,Location)$

R3: $nsubj(z,Bird) \wedge conj_and((z \vee Location,Location) \longrightarrow located_in(Bird,Location)$

Both rules R1 and R3 cover the negative example. Then the dependencies of negative examples are searched to find clauses which are not in the list of atoms relevant to positive examples. The rules are augmented with negation of the clause specific to negative example to uncover the negative example, but to still cover the all positive examples. Accordingly rules R1 and R3 will be modified as follows.

R4: $nsubj(z,Bird) \wedge prep_in(z,Location) \wedge \neg prep_except(z,Location) \longrightarrow located_in(Bird,Location)$

R5: $nsubj(z,Bird) \wedge conj_and((z \vee Location,Location) \wedge \neg prep_except(z,Location) \longrightarrow located_in(Bird,Location)$

Since a list of ordered atoms is created according to the number of occurrences on the positive examples, rules formed combining each atom from the list with the main atom

nsubj, cover positive examples. BeamSearch algorithm is developed to create rules from the list of atoms and modify each rule not to cover any negative examples. Then BeamSearch algorithm always gives the best rule and finally there will be minimum number of rules required to identify a relation.

Algorithm for ILP

Covering Algorithm

```

Create a list of atoms ordered according to the number of occurrences
(i.e. most occurred atom at the head and least occurred atom at the end)
Initialize the consequence of the rule  Consequence = Relation
Repeat
    Call the BeamSearch algorithm to find the best body BestBody
    For all training data in  $E^+$ 
        Apply the BestBody
        Remove the covered positive examples from  $E^+$ .
        Add the BestBody to the rule set
Until  $E^+ = \emptyset$ 

```

BeamSearch Algorithm

```

Initialize condition of the rule to Head_of_List
condition = condition & Head_of_Tail
Remove Head_of_Tail from the list
For all training data in  $E^-$ 
    Apply condition
    If a negative example is covered, add the complement of an atom specialized to the
    negative example to the condition to uncover the negative example but cover the
    positive examples.
    Add the clause  $\neg$ negative(verb) to the condition
BestBody = condition

```

Normally the initial maximally general rule body will be specialized by extending it with another atomic formula as shown in our example rule set. It is augmented with

counterfactuals of the atoms specific to negative examples to cover the positive examples but not to cover any negative examples. If there are no such atoms to be found, the main verb of the negative sentence is considered as a negative verb and the set of the negative verbs is updated with the verb found.

For e.g. the following two sentences give negative instances for the relation *located_in*, but both the sentences are covered by one rule in the initial rule set (above shown R3)

The sentence

Cranes live on all continents except Antarctica and South America

which indicates negative relations

\neg *located_in*(Crane Antarctica) and
 \neg *located_in*(Crane South America) gives reduced dependency

nsubj(live-2, Cranes-1)
prep_on(live-2, continents-5)
conj_and(Antarctica-7, South America-9)
prep_except(continents-5, South America-10)

From the above dependency prep_except can be considered as the specific atom for \neg *located_in* and it normally does not occur for *located_in*. Therefore when R3 is modified to form R5 by adding \neg prep_except to the rule body, this negative sentence will no longer be covered by a rule.

The sentence

Swans are absent from tropical Asia, Central America, northern South America and the entirety of Africa

which also gives negative relations gets the following dependencies from the Stanford parser.

nsubj(absent-3, Swans-1)
prep_from(absent-3, tropical Asia-6)
prep_from(absent-3, Central America-9)
conj_and(tropical Asia-6, Central America-9)
amod(America-13, northern-11)
prep_from(absent-3, America-13)
conj_and(Asia-6, America-13)
prep_from(absent-3, entirety-16)

conj_and(Asia-6, entirety-16)
 prep_of(entirety-16, Africa-18)

In the dependencies of this sentence there are no specific atoms to indicate the negation of the relation. Therefore the main verb “absent” which indicates the negation, will be added to the set of negative verbs for the relation. All the rules (R4, R2, R5) are also augmented with \neg negative(verb) and appropriate syntactic category is substituted for bound variable z in the atomic formulas. Then above rule set can be seen as follows

$$\begin{aligned}
 \text{R4'} : \quad & nsubj(verb, Bird) \wedge prep_in(verb, Location) \wedge \neg prep_except(verb, Location) \wedge \\
 & \neg negative(verb) \longrightarrow \\
 & \quad located_in(Bird, Location) \\
 \text{R2'} : \quad & nsubj(verb, Bird) \wedge prep_to(verb, Location) \wedge \neg negative(verb) \longrightarrow \\
 & \quad located_in(Bird, Location) \\
 \text{R5'} : \quad & nsubj(verb, Bird) \wedge conj_and((verb \vee Location), Location) \wedge \\
 & \neg prep_except(verb, Location) \wedge \neg negative(verb) \longrightarrow \\
 & \quad located_in(Bird, Location)
 \end{aligned}$$

Since the beam contains the elements for rule construction and not the rules, the size of the beam is not a significant factor that affects efficiency of the method explained here as in NEWGEM algorithm. In addition to that no rules become redundant in this method because of the heuristic approach taken in selecting atoms to form the rules.

4.4.7 Weakening the language bias

Language bias is the mechanism employed by a learning system to constrain the hypothesis space [34]. Normally in the implementation of LINUS the selected hypothesis language is restricted to constrained deductive hierarchical database (DHDB) clauses. In DHDB variables are typed and recursive predicate definitions are not allowed. In addition all the variables that appear in the body of a rule should appear in the head of the rule as well (i.e. relation clause in relation-extraction-rules). It was shown [34] that the language bias in LINUS can be weakened to include clauses which introduce new variables. The idea of determinacy allows for a restricted form of new variables to be introduced in the learned clauses. Determinacy of a predicate expression is defined as determinate if its clauses are determinate and clauses are determinate if each of its literals is determinate

[34]. A literal is determinate if each of its variables that do not appear in preceding literals has only one possible binding given the bindings of its variables that appear in preceding literals. Stanford dependencies hardly provide clauses with the instances of both entity types.[Appendix G] Therefore using Stanford dependency clauses as background we need to use additional variables in the rule body. These variables are generalized to syntactic categories and used inside the atomic expressions of the rule body. In the above relation-extraction-rule R2' the clauses in the rule body are determinate because each occurrence of new variable (i.e.verb in *nsubj(verb,Bird)*, *prep_in(verb,Place)* etc.) has only one possible binding given particular values of the other variables in the clause. Since a general term as a syntactic category is used as a variable in atomic expressions, language biasness of the algorithm is tend to be relaxed. Initially the system is biased by the length of concept representation in the rule body as restricted the number of atomic expression is restricted to two. But it is not necessary to carry out post processing in order to eliminate irrelevant clauses from the rule body to make the induced hypothesis more compact and accurate because two most relevant clauses are used at the initial step.

4.5 Markov Logic Network for Statistical Relation Extraction

MLN requires grounding all the first order clauses by substituting constants for all the variables in them. In the case of relation-extraction-rules verbs and entity instances in the training data corpus are used to ground the relation-extraction-rules. Since the number of grounding is intractable with large number of substitutions, reducing the number of clauses in the condition of the rules is vital for efficient implementation before MLN is used on them. Sets of negative and positive verbs are obtained during the implementation of ILP method for rule generation. Therefore negative verbs can be removed from the set of verbs which are used to ground the formulas. Then $\neg negative(VB)$ can be omitted from the rules because all the verbs used in MLN are positive verbs. Reduction of clauses in the condition part of the rules is explained using the following example set of rules which are the generalized first order form of above deduced rules in the section 4.4.6 for the relation *located_in(Bird,Location)*. Relation-extraction-rules are given in Appendix H.

$$\begin{aligned}
& \forall x \forall y ((nsubj(VB, x) \wedge prep_in(VB, y) \wedge \neg negative(VB) \wedge \neg neg(VB, not)) \longrightarrow located_in(x, y) \text{ ————— } (1) \\
& \forall x \forall y \exists z ((nsubj(VB, x) \wedge conj_and((VB \vee z), y) \wedge \neg prep_from(VB, y) \wedge \neg prep_for(VB, y) \wedge \neg prep_except((NN, y) \\
& \quad \wedge \neg negative(VB) \wedge \neg neg(VB, not)) \longrightarrow located_in(x, y) \text{ ————— } (2) \\
& \forall x \forall y \exists z ((nsubj(VB, x) \wedge conj_and(z, y) \wedge \neg prep_from(VB, z) \wedge \neg prep_for(VB, z) \wedge \neg prep_except((NN, z) \\
& \quad \wedge \neg negative(VB) \wedge \neg neg(VB, not)) \longrightarrow located_in(x, z) \text{ ————— } (3) \\
& \forall x \forall y ((nsubj(VB, x) \wedge prep_on(VB, y) \wedge \neg negative(VB) \wedge \neg neg(VB, not)) \longrightarrow located_in(x, y) \text{ ————— } (4) \\
& \forall x \forall y ((nsubj(VB, x) \wedge prep_to(VB, y) \wedge \neg negative(VB) \wedge \neg neg(VB, not)) \longrightarrow located_in(x, y) \text{ ————— } (5)
\end{aligned}$$

Where VB = Verb, NN = Noun, $x \in Bird$ and $y, z \in Location$

The atom $\neg neg(VB, not)$ is relevant to a particular pair of *Bird* and *Location* instances. But the atom itself does not contain *Bird* or *Location* variables because the rules normally applied to the reduced dependencies of a sentence. For an example if $neg(found, not)$ is found in one training example $\neg neg(found, not)$ will be considered as an evidence atom in MLN leading to inaccurate truth assignment for formulas. Therefore $\neg neg(VB, not)$ is also be omitted from rules in MLN modeling. The negative literals $\neg prep_for(VB, z)$ and $\neg prep_except((NN, z)$ in rules (2) and (3) have the same impact and therefore can also be removed from the rules. But $\neg prep_from(VB, z)$ directly associates with a negative verb. Since negative verbs are not used in grounding atoms $\neg prep_from(VB, z)$ can also be ignored. Although $nsubj(VB, x)$ is also common to all the rules it cannot be treated the same way as negative clauses because it contains one of the bound variables(i.e. x) which comes in the clause to be inferred(i.e. the relation clause). Therefore first and second clauses should be in conjunction to make the rules meaningful for possible relation inferences. Then the rule set which is used for MLN will be as follows.

$$\begin{aligned}
& \forall x \forall y ((nsubj(VB, x) \wedge prep_in(VB, y) \longrightarrow located_in(x, y) \text{ ————— } (1)' \\
& \forall x \forall y ((nsubj(VB, x) \wedge conj_and((VB, y) \longrightarrow located_in(x, y) \text{ ————— } (2)' \\
& \forall x \forall y \exists z ((nsubj(VB, x) \wedge conj_and(z, y) \longrightarrow located_in(x, z) \text{ ————— } (3)' \\
& \forall x \forall y ((nsubj(VB, x) \wedge prep_on(VB, y) \longrightarrow located_in(x, y) \text{ ————— } (4)' \\
& \forall x \forall y ((nsubj(VB, x) \wedge prep_to(VB, y) \longrightarrow located_in(x, y) \text{ ————— } (5)'
\end{aligned}$$

4.5.1 Applicability of MLN on Relation-Extraction-Rules

Identified entities and verbs from the dependencies are used in grounding the relation-extraction-rules when modeling them in MLN. In addition the knowledge base consists of evidence which are considered as known atoms because truth value of evidence atoms are known. Evidence is required in order to find the truth value of a formula at a particular state. Modeling extraction rules in MLN is explained here with respect to rule (1)

mentioned above. Three example sets which contain two members in each and taken from the training data are used to ground the rule. All the possible groundings of the rule and the evidence (available groundings in the training data set) are shown in table 4.5.

Rule

$$\forall x \forall y ((nsubj(VB, x) \wedge prep_in(VB, y)) \longrightarrow located_in(x, y))$$

Constants from 3 sets

VB \in Verb = {are_native, found}

x \in Bird = {Ostrich, Parrot}

y \in Location = {Australia, Africa}

Possible Groundings			Evidence		
nsubj(VB, x)	prep_in(VB, y)	located_in(x, y)	nsubj(VB, x)	prep_in(VB, y)	located_in(x, y)
nsubj(are_native, Ostrich)	prep_in(are_native, Australia)	located(Ostrich, Africa)	nsubj(are_native, Ostrich)	prep_in(found, Australia)	located_in(Ostrich, Africa)
nsubj(found, Ostrich)	prep_in(are_native, Africa)	located(Ostrich, Australia)	nsubj(found, Parrot)		located_in(Ostrich, Africa)
nsubj(are_native, Parrot)	prep_in(found, Australia)	located(Parrot, Africa)			
nsubj(found, Parrot)	prep_in(found, Africa)	located(Parrot, Australia)			

Table 4.5 Possible groundings and Evidence with respect to Rule(1) and the constants

The resulting network of the rule with the possible groundings are shown in the figure 4.10.

Algorithm for the construction of all the groundings with respect to relation-extraction-rules

F- a set of an extraction rules

E1- a set of instances of entity1

E2 – a set of instances of entity2

VB – a set of verbs

NN- a set of nouns

NVB – a set of negative verbs

G_F - set of ground atoms

G_F = { \emptyset }

For each rule in F

Convert the rule into its Clausal form CNF(F)

If NBV $\neq \emptyset$

Remove the negative verbs from the VB (PVB \leftarrow VB \ NVB)

F \leftarrow CNF(F)

For each clause F_i \in F

G_i = {F_i}

For each variable x in F_i

For each clause F_j(x)

If the type of x is entity1

Obtain the ground clauses substituting all the values from E1

If the type of x is entity2
 Obtain the ground clauses substituting all the values from E2
 If the type of x is VB
 Obtain the ground clauses substituting all the values from PVB
 If the type of x is NN
 Obtain the ground clauses substituting all the values from NN
 $G_i \leftarrow (G_i \setminus F_i(x)) \cup \{F_i(c_1), F_i(c_2), F_i(c_3), \dots\}$
 (where c_1, c_2, c_3, \dots represent the members of E1, E2, PVB, NN)
 $G_f \leftarrow G_f \cup G_i$

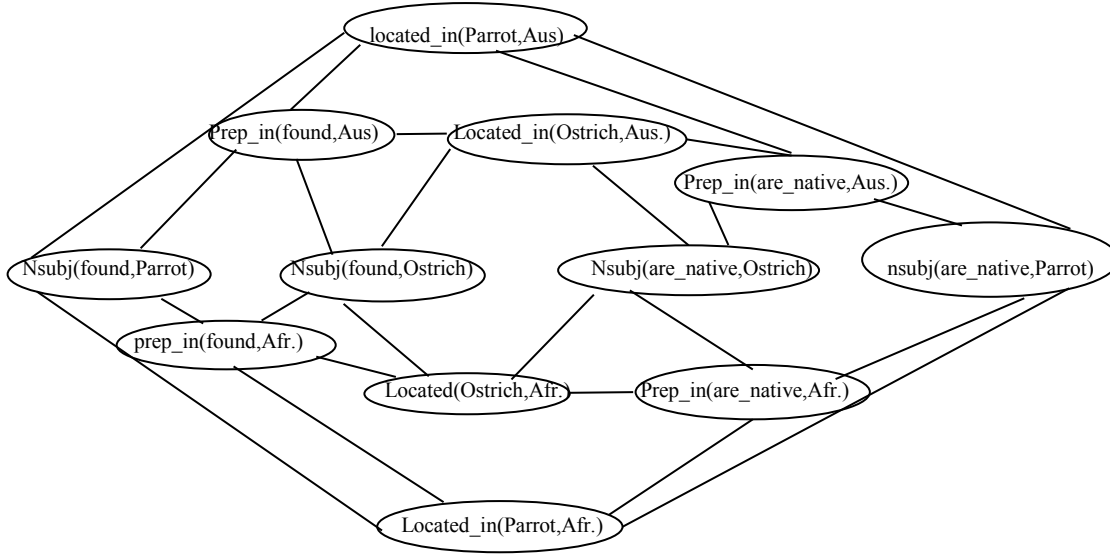


Figure 4.10 The Network of the grounded atoms with respect to rule 1

4.5.2 Sampling Atoms for MLN

MLN require counting the number of true groundings of a formula at a given world state. The probabilistic state space created by a large data base is intractable to do these counting. The higher the number of objects in the MLN the more difficult the computations become. With even rather small training set that is used in ILP there are 960 grounded atoms with respect to the set of rules given at the beginning of the section 4.5 for the relation *located_in()*. In this situation the state space can be reduced by removing the known true literals from the MLN and removing the negative verbs from the set of verbs.

Since there is a set of equivalent verbs for the relation verb generated during the ILP process it is possible to replace an equivalent verb with the main relation verb. Then most probably there is only one verb for each relation. In this way number of atoms in the

initial MLN can approximately be reduced to a number which fluctuates around half of the initial number of atoms, depending on the number of evidence atoms available. Figure 4.11 shows the MLN of the above rule when the evidence atoms are removed and figure 4.12 shows the further reduced MLN by replacing all the equivalent verbs with the relation verb. Further the availability of negative training data can be used to eliminate negative relation clauses from the MLN.

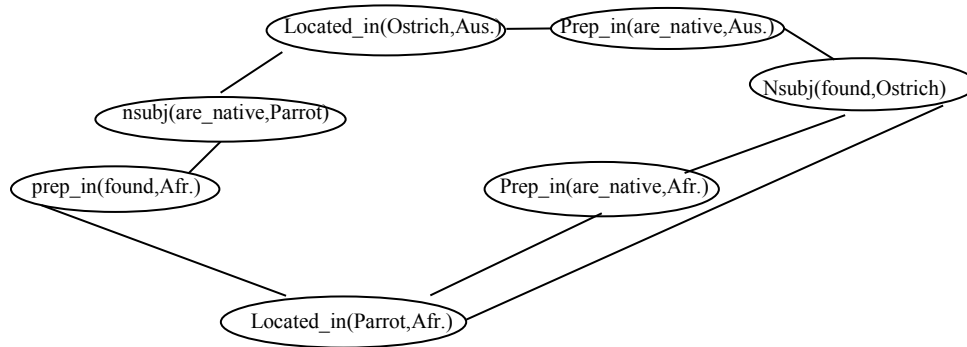


Figure 4.11 Reduced Network after removal of the evidence

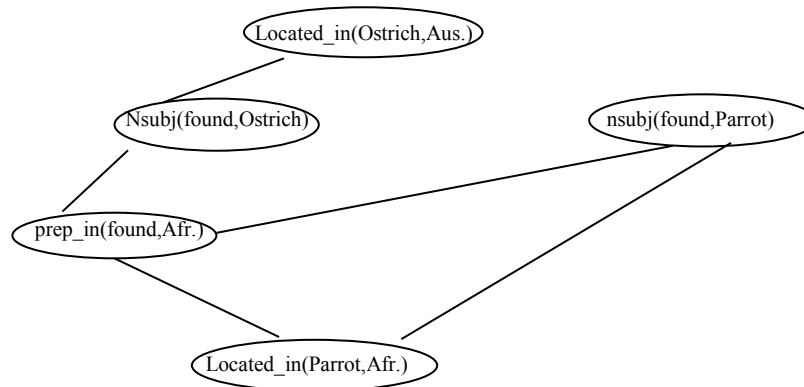


Figure 4.12 The network when the verbs are replaced with one equivalent verb.

4.5.3 Weight Learning for Relation-Extraction-Rules

Weights of first order formula can be learnt generatively or discriminatively. Weights can be calculated generatively by maximizing a likelihood or pseudo-likelihood of a relational database. Since the computations in generative learning is highly intractable and as in many applications as in system explained in here and the pseudo-likelihood parameters may lead to poor results when inference across non-neighbouring variables is required, discriminative learning [33] is preferred in weight learning for relation-extraction-rules. In addition to that a priori which predicates will be evidence and which will be queried is known, makes discriminative learning [33] more suitable for the purpose. In discriminative learning conditional likelihood of query atoms is used. The conditional likelihood of query atoms Y given evidence atoms X is given by (5)

$$P(y | x) = (1 / Z_x) \exp(\sum_{i \in F_y} w_i n_i(x, y)) \quad (5)$$

Where F_y is the set of all MLN clauses with at least one grounding involving a query atom and $n_i(x, y)$ is the number of true groundings of the i^{th} clause involving query atoms. The gradient of the Conditional log-likelihood is given by

$$\begin{aligned} \partial / \partial w_i (\log P_W(y | x)) &= n_i(x, y) - \sum_{y'} P_W(y' | x) n_i(x, y') \\ \partial / \partial w_i (\log P_W(y | x)) &= n_i(x, y) - E_W[n_i(x, y)] \end{aligned} \quad (6)$$

Although the number of grounded atoms can be reduced as explained above, computing expected counts E_W is intractable. Closed World Assumption cannot be used with the dependency literals because the domain is infinite though limited number of training data is used in the experiment. Therefore E_W can be approximated by the counts $n_i(x, y_w^*)$ in the MAP(Maximum A Posteriori) state. In the problem domain given under experimental results, finding single MAP state is not guaranteed because same conditional probability value exists for number of states. Therefore Contrastive Divergence (CD) [86] is used in gradient calculations instead of using MAP state. CD approximates the expectations from a small number of Monte Carlo Markov Chain (MCMC) samples. Gibbs sampling is chosen with CD in order to create samples of states. In using Gibbs sampling random numbers are used in assigning truth values for atoms from conditional probability. The conditional probability of each ground atom within its Markov Blanket is used for Gibbs sampling. Each Gibbs step consists of sampling a ground atom when its Markov blanket

is given. Gibbs sampling requires weights of rules in its sampling process. The weight of a rule is calculated basically for Gibbs sampling by the log odds between a world where the rule is true and a world where the rule is false when other things are equal. But this phenomenon cannot be applied to find the actual weight of a relation-extraction-rule because rules in system share variables with each other. However the weights calculated in this manner is used only for the sampling. Weight is calculated for each Markov Blanket separately.

The Markov blanket of a ground atom is the set of ground atoms which has direct links with it in MLN. Markov blanket of $nsubj(found, Ostrich)$ with regard to example knowledge base and the rule is shown in figure 4.13. Then the probability of a ground atom X_i with respect to a Markov Blanket B_i is given by

$$p(X_i = x_i | B_i = b_i) = \frac{\exp(\sum_{f_i \in F_i} w_i f_i(X_i = x_i, B_i = b_i))}{\exp(\sum_{f_i \in F_i} w_i f_i(X_i = 0, B_i = b_i)) + \exp(\sum_{f_i \in F_i} w_i f_i(X_i = 1, B_i = b_i))} \quad (7)$$

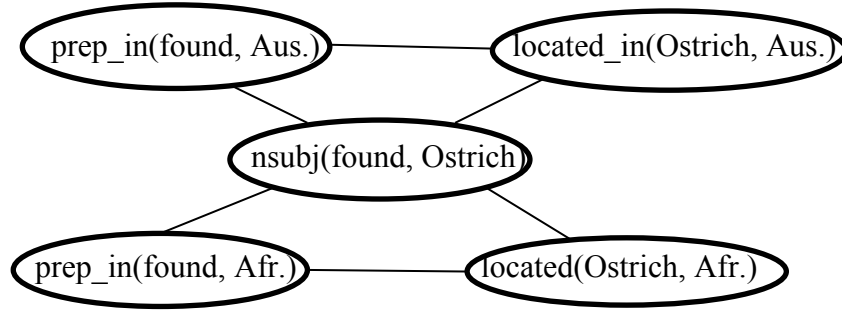


Figure 4.13 Markov Blanket of the atom $nsubj(found, Ostrich)$

Table 4.6 shows the state space of the Markov blanket of $nsubj(found, Ostrich)$ with possible truth values

located_in(Ostrich,Afr.)	prep_in(found,Aus)	nsubj(found,Ostrich)	prep_in(found,Afr.)	located_in(Ostrich,Aus.)	n_i	$P(x_i)$
1	1	1	1	1	2	e^{2w}
1	1	1	1	0	1	e^w
1	1	1	0	1	2	e^{2w}
1	1	1	0	0	1	e^w
1	1	0	1	1	2	e^{2w}
1	1	0	1	0	2	e^{2w}
1	1	0	0	1	2	e^{2w}
1	1	0	0	0	2	e^{2w}

Table 4.6 State space of the Markov blanket of $nsubj(found, Ostrich)$

Weight calculated for the rule 1 on this truth table is 0.85

Algorithm for Gibbs Sampling from the clausal form of Relation-extraction-rules

Ev - set of evidence atoms

Nev - set of non-evidence atoms

Repeat

For each $a_i \in \text{Nev}$

Find the Markov Blanket

Create the state space of the Markov blanket

Assign truth value for a_i (0 or 1)

Pick a random state with the assigned value of a_i

Calculate the log odd weight

Find the probability of the state $P(x_i)$ according to equation (5)

Generate a random number r_i between 0 and 1

If $P(x_i) > r_i$

Change the truth assignment of a_i

Else

Sample atom a_i

Until all the atoms are sampled.

4.5.4 Weight Optimization

Equation (6) poses a multivariate weight optimization problem. Gradient Descent, Diagonal Newton and Conjugate Gradient are available multivariate optimization techniques for efficient weight learning for MLN [86]. Gradient Descent is comparatively slow and Diagonal Newton has limitations in uncorrelated clauses. Therefore Conjugate Gradient method is preferred for weight optimization in the experiment. In Conjugate Gradient method search directions are constructed by conjugation of residuals and Polak-Ribiere method [87] is used to find conjugate direction though there are several equivalent expressions for this. Polak-Ribiere method often converges much more quickly.

Algorithm for Conjugate Gradient Method

Choose initial weight vector w_1

Set $p_1 = r_1 = -f'(w_1)$, $k = 1$

While $r \neq 0$

Calculate second order information

$$s_k = f''(w_k)p_k \quad \delta_k = p_k^T s_k$$

Calculate step size α

$$\mu_k = p_k^T r_k \quad \alpha_k = \mu_k / \delta_k$$

Update weight vector

$$w_{k+1} = w_k + \alpha_k p_k \quad r_{k+1} = -f'(w_{k+1})$$

If $k \bmod N = 0$ then restart algorithm

N-Number of iterations set
for the system

$$p_{k+1} = r_{k+1}$$

Else

Create new conjugate direction

$$\beta_k = (r_{k+1}^T (r_{k+1} - r_k)) / r_k^T r_k$$

$$p_k = r_{k+1} + \beta_k p_k$$

$k = k+1$

In scaled conjugate gradient method the complexity of calculating hessian matrix is reduced by the substitution

$$s_k = f''(w_k)p_k \approx (f'(w_k + \sigma_k p_k) - f'(w_k)) / \sigma_k \quad \text{where } 0 < \sigma_k \ll 1$$

When $\delta_k < 0$ (if $\delta_k < 0$ then hessian is not positive definite and algorithm will not work)

$$s_k \approx (f'(w_k + \sigma_k p_k) - f'(w_k)) / \sigma_k + \lambda_k p_k$$

It is needed to adjust λ_k in each iteration looking at the sign of δ_k in order to make it positive.

$$\text{Then } s_{knew} = s_k + (\lambda_{knew} - \lambda_k)p_k$$

Condition $\lambda_{knew} > \lambda_k - \delta_k / |p_k|^2$

$$\lambda_{knew} = 2(\lambda_k - \delta_k / |p_k|^2)$$

Scaled Conjugate Algorithm

Choose initial weight vector w_1 and scalars $0 < \sigma < 10^{-4}$, $0 < \lambda_1 < 10^{-6}$, $\lambda_{knew} = 0$

Set $p_1 = r_1 = -f'(w_1)$, $k = 1$ and success = true

While $r \neq 0$

If success is true then calculate second order information

$$\sigma_k = \sigma / |p_k|, \quad s_k = (f'(w_k + \sigma_k p_k) - f'(w_k)) / \sigma_k, \quad \delta_k = p_k^T s_k$$

Scale δ_k $\delta_k = \delta_k + (\lambda_k - \lambda_{knew})|p_k|^2$

If $\delta_k < 0$ then make the Hessian matrix positive definite

$$\lambda_{knew} = 2(\lambda_k - \delta_k / |p_k|^2) \quad \delta_k = -\delta_k + \lambda_k |p_k|^2 \quad \lambda_k = \lambda_{knew}$$

Calculate step size α

$$\mu_k = p_k^T r_k \quad \alpha_k = \mu_k / \delta_k$$

Calculate the comparison parameter

$$\Delta_k = 2\delta_k [f(w_k) - f(w_k + \sigma_k p_k)] / \mu_k^2$$

If $\Delta_k \geq 0$ then a successful reduction in error can be made

Finally the probabilities of the existence of the extracted relation instances are calculated according to equation (4). The denominator of the equation contains the probability of evidence atoms. Overview of the weight learning process is shown in figure 4.14.

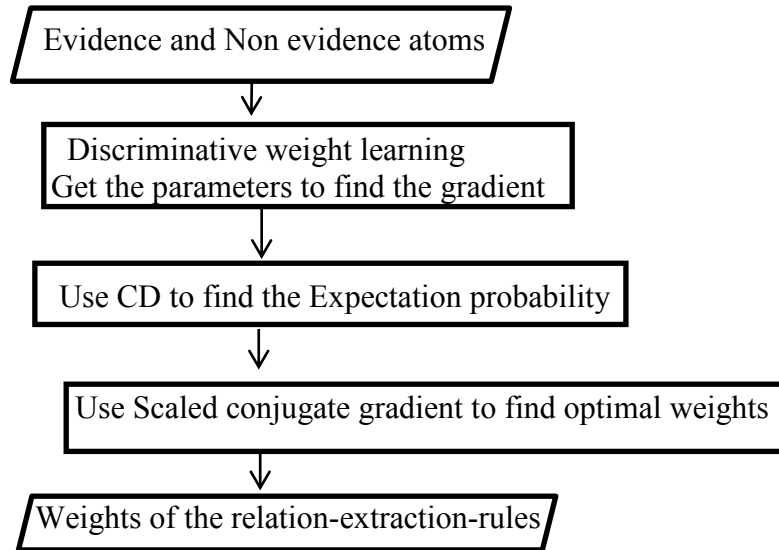


Figure 4.14 Overview of the weight learning process

Chapter 5

Use of Relation-Extraction-Rules on Document Classification

5.1 Introduction

The importance of document classification is mentioned in chapter 1 and state-of-the-art methods and newly developed methods for document classification are discussed in chapter 3 under Related Work. Information extraction methodology developed here has a great potential to be used for document classification and it has an added advantage that both information extraction and document classification can be performed simultaneously. Further, document classification becomes an application of information extraction. .

Section 5.2 presents the document representation for the proposed method and the proposed method for document classification.

Extensive comparison of the present method with the state-of-art document classification methods and newly developed methods is given in the section 5.3.

5.2 Document Classification

In document classification, the success of the classification method depends on the document representation. Therefore as mentioned in the previous chapters finding the most suitable set of features is a challenging task in text classification. Normally conventional document representations based on class specific word statistics contains irrelevant noisy features which makes feature set unnecessarily long. Therefore some recent work have made effort to address the issue of finding the optimal feature set by developing methods for discriminating feature selection and adopting a localized feature selection approach [78, 79].

Domain specific entities and associated relations can be considered as good candidates for class specific features. Then features to represent documents can be reduced to class specific entities and relations and a class of a document can be defined by a set of predefined domain specific entities and associated relations as similar to the other bag of

words approaches. This implies that word based document representation can be replaced by information based representation.

5.2.1 Representing Documents in Entity-Relation framework

Under the information based document representation, class C_i can be modeled by a set of entities Ec_i and a set of entity relation predicates Rc_i which are embedded in *relation-extraction-rules*, as in (1) and (2).

$$C_i :- Ec_i = \{e_1, e_2, \dots, e_n\} \quad (1)$$

$$\exists e_i, e_j \in Ec_i = \{r_1(e_i, e_j), \dots, r_m(e_i, e_j)\} \quad (2)$$

A document D_t contains a number of classified relations between identified entities that are specific to its domain and the entities and relations can be derived from document D_t as shown in (3).

$$D_t \vdash Ed_t, Rd_t \quad (3)$$

where Ed_t (set of entities in D_t) $\subseteq Ec_i$ and Rd_t (set of relations in D_t) $\subseteq Rc_i$

For an example Table 5.1 shows some examples for relations and respective entity tuples for two domains. Pre-defining entities and relations for document classes depends on the application of the text classification and user community. Entities and Relations shown in Table 5.1 are biased towards general purpose information extraction. First the documents are passed through the entity extraction phase in order to identify the class specific entities present in the documents. Then the relation-extraction-rules are used on the documents annotated with entities to find the relations existing between entities.

A document might not contain all the entities and relations assigned to a class. Therefore a subset with cardinality beyond a threshold value which is determined based on the training corpus can be accepted.

5.2.2 Determination of Classification Performance

Under a rule based classification approach a set of rules is extracted from training data. The antecedent of the rule contains the condition which relies on the feature set while consequent defines the possible class label. Normally the condition consists of a pattern of word combinations, presence of terms and a large number of such rules are generated

for a predefined class. But the rule based methods suffer from irrelevant noisy features and large number of rules. Two of most commonly used criteria to use in rule generation are those of support and confidence [81]. Support indicates the number of instances in the training set which are relevant to the rule and Confidence is the conditional probability that an instance in the training set belongs to a class given by the rule when the condition is satisfied. However support does not give clear indication of the strength of the rule whereas Confidence is more direct basic measure of the rule strength. But Support and Confidence are the most used measures in ordering and refining the rule set.

Rule based classification is preferred in most practical scenarios because of its ease of maintenance and interpretability. When a test instance satisfies a number of rules with the same class label at the condition of the rule, that class can easily be assigned to the test instance. But when the rules are relevant to different classes the above mentioned confidence measure is used for conflict resolution. RIPPER is one of the most common techniques used in rule generation which determines the frequent combinations of words relevant to a class. The technique Sleeping Expert finds sparse phrases which are groups of neighboring words (not necessarily sequential) to be used in weighted rules. Since the measures Support and Confidence do not normalize for a prior presence of different terms and features, the classification rules are prone to misinterpretation on training data corpus with imbalanced class distribution. When a document class is signified by a large number of rules, confidence based conflict resolution might not be sufficient for accurate classifications. This emphasizes the requirement of more sophisticated techniques for conflict resolution. Weight is learnt for each relation-extraction-rule as explained in chapter 7 and rule weights are used in measuring the strength of classification.

Therefore the measure Class Index (CI) is defined based on rule weights, to determine the appropriateness of assigning a class for a document when the relation-extraction-rules from different classes or domains are applied during the classification process. CI is calculated in terms of weights on the basis of number of rules applicable on the document and it is shown in (4)

$$CI = \sum_i w_i I(r_i) / \arg \max_i \sum_i w_i \quad (4) \quad \text{Where } w_i \text{ is weight of the } i^{th} \text{ rule. } I(r_i) \text{ is the indicator function which has the value 1 if the rule is applicable on the}$$

document else $I(r_i)$ is 0. CI gives a clear indication of the number of relations found in a document and the strength of the relation-extraction-rules applied on the document.

Since rules are generated for each class independently, the imbalanced class distribution in the document collection will not have an adverse effect on the classification process.

Class	Relation	Entity Tuple
Bird	Located_in	Bird, Location
	Eat	Bird, Diet
	Has_characteristic	Bird, Bird_Part
	Related	Bird, Bird
	Nest_in	Bird, Nest
	Has_length	Bird, length
	Has_weight	Bird, weight
	Lay_eggs	Bird, Egg_number
	Is_a	Bird, Super_bird
Sport	Play_with	Sport, Tool
	Play_by	Tool, Action
	Made_of	Tool, Material
	Has_player	Sport, Player_number
	Has_length	Tool, Length
	Has_width	Tool, Width
	Has_weight	Tool, Weight
	Played_in	Sport, Location
	Is_a	Sport, Super_sport

Table 5.1 Examples of relations and respective entity tuples

5.3 Comparison of the Proposed Document Classification Method with other Related Work

First the proposed approach is discussed in line with well-established popular methods of text classification and some other approaches discussed under Related Work in Chapter 3. Then it is compared with a similar classification method [80] which is also based on information extraction. Since proposed text classification system is completely based on entity and relation extraction it is finally reviewed at the point of information extraction's view with recently published work [59] in that area.

The text classification methods such as Naive Bases [64, 81] Support vector machines [67, 68], Centroid [73], Rocchio [81] and K-nearest neighbour [72] use bag of words representation for documents. Bag of words representation can contain thousands of different words in the document vector and there will be a considerable number of

irrelevant words with respect to the document class. The expensive computations on both training and classification phases affect the performance efficiency adversely. In the proposed method bag of words document representation is replaced by entity relation tuples. The number of Entities and Relations in a text document is much less than the number of different words found in a document. Entities and their relationships are defined for a class depending on the application and they all are relevant features for a class. Therefore in the proposed method there is no issue of irrelevant noisy features coming into document representation. Relation extraction rules capture the correlation of individual words through dependencies which address the issue of poor classification due to independence assumption in Naive Bases classification method. Since a training corpus is considered to be belonged to one class at a time to generate relation-extraction-rules, the proposed method is completely independent of the class distribution of the training corpus whereas above mentioned other methods directly or indirectly depends on the class distribution in the training corpus. Especially in naive based classification class prior parameter, calculation of which depends on the class distribution of the training document set directly involves in probability calculations. Therefore Naive based method can lead to inaccurate probabilities resulting in incorrect classification when class distribution of the training corpus is skewed.

In Support vector machines, Centroid, Rocchio and K-nearest neighbour methods, a generation of acceptable linear classifier is vital in accurate text classification. But generation of good linear classifier is not guaranteed in any of the methods which totally depend on the training corpus. Since we consider entities and relations specific to a class and each relation extraction rule binds class specific entities, only a few number of overlapping can be expected in the proposed method. Hence a good classification can be achieved when the *relation-extraction-rules* are applicable on the test documents. CenKNN [74] which is proposed recently to address the drawbacks of the individual methods Centroid and K-nearest neighbour, accomplishes it by reducing the dimensions in the document representation at the expense of computational cost. Although the dimension of document representation is reduced to the number of classes in the training corpus a new cost is incurred in computations in dimension reduction process, affecting the efficiency of whole process. An acceptable classification is achieved by generalized

representation(GI) in GIS method [75] that uses KNN. But as mentioned in section 2, other drawbacks of the KNN method except skewed class distribution are not addressed. In proposed method there aren't complicated computations except in weight learning process for *relation-extraction-rules*. Therefore there are provisions to update training corpus without disturbing the entire system. New rules can simply be added to the rule base when the training corpus is updated. Then it is a matter of finding optimal weights for the rules. But once the optimal weights for rules are found the system will not be disturbed in any way until the rule base is modified by new additions. Any way in that case classification phase is not affected and modifications are done only in the training phase.

In most rule based systems [78, 79, 82] the conditions in the rules are mere combination of words taken from training documents despite of the fact that attempts [78, 79] have been made to prune the number of rules and the components in the antecedent of the rules at the expense of the classification accuracy. Therefore in employing these pruning techniques a loss of relevant information can be expected. But in the condition of the *relation-extraction-rules* consists of dependency clauses which are obtained from relevant minimized dependencies of sentences, to identify relation instances and clauses to prevent extraction of false relation instances. Therefore in *relation-extraction-rules* there are only two clauses to extract relations and maximum of five other clauses for correct identification of relations instances when compared to large number of components in other rule based systems.

The technique (explained in detail in the Related Work in Chapter 3) employed in classification method based on information extraction presented by Riloff et al [80] which is similar to the proposed method, relies heavily on domain specific dictionary of concept nodes. Although the three algorithms explained use varying amount of extracted information to classify texts to achieve a high precision the recall is average or less. Adding more information to the algorithm relevance signature which is with least amount of information will make the method more specific and may miss highly relevant text when there are no specific words or phrases to capture relevancy, resulting low recall. Augmented Signature Algorithm and case based algorithm try to extract information to combine the keyword or phrases with context in which they appear. In the proposed

method the contexts are tackled with domain specific entities and their relations identifying correlation between individual entities. The entities are the keywords or phrases and relations between entities which are captured through dependencies of individual sentences, explains the context within which the entities exist. Since the natural language sentences come in various forms and can be unnecessarily long with irrelevant words the dependencies of sentences are processed to filter out unnecessary words. Therefore dependencies are reduced in both training and text sentences to capture underlying semantic information wrapped in the sentence. Concept node may not be able to instantiate some relevant information in the free text because there is a fair chance that the concept node framework may overlook the information in unprocessed sentences.

All three algorithms discussed in Riloff et al's publication are based on the concept node definitions. Any relevant information not triggered by concept node dictionary will be unaccounted in the classification process. On the other hand there are more than one definition for same trigger word depending on syntactic existence of a word in the text. Then even the active-passive nature is addressed by two different concept nodes for same trigger word leading to two probability values for a text context of same nature. This will adversely affect the accuracy of probability calculations and hence the classification. The proposed method generates number of different rules for the same relation. But it does not consider active or passive voice sentences differently and collapse all the equivalent relation verbs in to a single relation. The ILP system which generates *relation-extraction-rules* might create rules for both active and passive sentences of a relation separately depending on the nature of the relation, not necessarily for all the relations.

Converting whole document into cases analyzing each sentence separately is not very efficient with long documents even with 100s of sentences when most of the sentences are not relevant. Although it may work on very specific piece of text it can be expected to work poorly on general purpose text. Most of extracted cases may not contain useful information whereas class specific entities and relations are very useful items in information extraction. Therefore both document classification and information extraction take place simultaneously. Number of entities and relations present in a document is much smaller than number of cases created from each sentence because all the sentences in the document do not contain entities. If there are no entities identified in

a sentence then there are no relations present. Therefore use of entities and relations to represent documents is more simple and efficient in the classification phase.

Method	Document Representation	Dimensionality	Cost of Computation	Accuracy	Applicability
Naive Bases	Bag of words	High	High	Depends on the Irrelevant term in the representation and the class distribution	Best for short documents
Support vector machines	Bag of words	High	High	Depends on the generation of good linear classifier.	High
Centroid, Rocchio and K-nearest neighbour methods	Bag of words	High	High	Depends on the generation of good linear classifier and the class distribution	Best for short documents
CenKNN	Bag of words	Low	High with an additional cost in the dimension reduction	Improvement due to reduction of the number of irrelevant terms	High
GIS Method	Bag of words	Moderate	High	Independent of the class distribution	Moderate
Rule based methods		Depends on the number of rules and the number of components coming to rules	Moderate	Depends on the applicability of the rules	High
Information Extraction method	Signatures created by extracted information	Low		High precision & low recall	Best for short documents
Relation-extraction-rules (Proposed method)	Class specific entities and relations	low	Low (high only in training phase)	High precision & recall depending on the applicability of rules	High

Table 5.2 Summarization of the comparison of the proposed approach with other related method

Since proposed document classification system builds on class-specific entities, entity extraction plays the most important role in the accurate classification. In using GATE to

extract entity instances semantic gazetteers and rules which incorporate patterns around annotated entity instance in a sentence are used. But this may not be possible with very specific entities which can be defined by descriptive noun phrases. The semantic parsing methods proposed by Choi et al [58] and Yih et al [83] will be more suitable for capturing such specific entities. But use of convolutional neural networks as in Yih et al method for semantic parsing is computationally expensive for entity extraction unless the entities are very uncommon and subjective for the application. Similarly in Choi et al method of matching under specified logical form of a noun phrase with a Freebase query can expect to be a lengthy process especially when there is no appropriate concepts match in the target ontology in the Freebase. Although expensive and complicated semantic parsing process is not feasible in extracting predefined general entities It can make the accurate entity extraction possible for any kind of domain making proposed method more comprehensive in document classification.

Summarization of this comparison of proposed document classification method with other related work is given in table 5.2.

5.4 Expanding the Training Text Corpus

When relation extraction rules are used on the reduced dependencies of a natural language sentence, one of the five types of outcomes can be expected. The three categories of ambiguous sentences with respect to relation-extraction-rules mentioned in chapter 4 section 4.4.3 are also included here. Here all kind of sentences are considered. Then any sentence can be categorized into one of the following five groups with respect to relation-extraction- rules.

- (i) Rules can cover the reduced typed dependencies of the sentence and extract relation instances from the sentence with a known verb(i.e. equivalent verb).
- (ii) Rules can cover the reduced dependencies of the sentence and extract relation instances with an unknown verb.
- (iii) Rules cannot cover the reduced dependencies of the sentence but the verb which binds the two entity instances in the sentence is known.
- (iv) Rules cannot cover reduced dependencies of the sentence and the verb is unknown.

(v) Rules cannot cover reduced dependencies of the sentence because the sentence is negative for the relation.

In the case of the sentences in category (i) relation instances are successfully identified with a high probability that the identified relation instances are true. The text corpus can simply be extended with the relevant details of the sentence.

Category (ii) sentences give the relation instances with less probability of them being true and with a possibility of updating the set of equivalent verbs for the relation with the new verb found.

With the category (iii) sentences there is a possibility of creating a new rule for the relation and updating the set of relation extraction rules with the new rule and text corpus with sentence details.

Category (iv) is completely uncertain. The sentence may give a new rule and an equivalent verb for the relation in question or give completely new relation. A manual involvement might be required in order to identify the relation in a case where the other methods fail to identify the situation correctly.

In the case of category (v) sentences the corpus can be updated with the sentence details as negative sentences for the relation.

However there is a considerable level of uncertainty involved with the outcomes (ii), (iii) and (iv). Therefore a method is proposed using characteristics which are not incorporated into the extraction rules, of the typed dependencies and frequencies of relation verbs along with their existential features in the database to estimate the validity of a sentence for the training corpus. The consistency of the extracted relation instances is tested to see how well they fit with the existing text corpus. Chi-square goodness of fit test can be used for this purpose on a Relation Index Value as explained below.

There are desirable features and undesirable features in a sentence annotated for relation extraction for a particular relation (i.e. positive and negative feature with respect to the relation). Those features and value of each feature can be identified to add a measure named Relation Index (RI). RI is calculated on reduced typed dependencies of a collection of sentences in all five categories mentioned above.

$$\text{Relation Index} = \sum_i f_i \quad \text{where } f_i \text{ is a feature}$$

Twelve Features which contribute to Relation Index have been identified and they are given below.

1. Verb Popularity (VP)

Verb popularity indicates the frequency of relation verb in the collection of sentences.

Where S – Sentence, S_v – sentence with the relation verb

2. Rule Weight(Highest_weight)

W is the weight of the rule which has the highest weight.

3. Distance Index (DI)

The distance information in the typed dependencies is made use here to find how far away the entities are in the sentence. DI is taken as the reciprocal of the distance between the two entities ($DI=1/(\text{distance between entity1 and entity2})$). When there are more than one entity instance pair in the sentence the minimum distance is considered.

4. Adverse Adverbs (AA)

AA indicates the presence of adverbs which affect the relation verb adversely in the sentence. AA is a negative feature. If there are adverse adverbs present in the sentence AA is assigned to -1 else it is assigned to 0. For a negative sentence it is always 0.

5. Dependency Consistency (DC)

The number of typed dependencies which contain the reduced typed dependency of the new sentence is calculated as a fraction of the total sentences in the collection.

6. Dependency Popularity (DP)

DP indicates frequencies of each dependency atoms in a sentence in the collection. DP is calculated as same as VP for each atom in the typed dependencies of the sentence separately .

7. Multiple Relationship (MR)

MR is a negative feature. If the dependencies are covered by the rules of more than one relation MR is assigned to -1 else it is assigned to 0.

8. Negative Relation Verb (NRV)

If the sentence has a negative verb NRV is assigned to -1 else it is assigned to 0.

9. Negative Dependency (ND)

Any dependency clause which indicates a negative relationship between two entities with respect to a relation is considered as a negative dependency clause. The presence of any negative dependency clause in the reduced typed dependency of a sentence makes ND to be assigned to -1 else ND is 0.

10. Verb Tense (VT)

VT is also a negative feature which is assigned to -1 if the main verb (verb in the nsubj clause) is a past tense verb. Otherwise VT is zero.

11. Verb Index (VI)

A sentence may contain more than one verb and the relation verb may be one of them. VI indicates the relation verb as a fraction of the number of verbs in the sentence.

12. Number of Relation Instances (NRI)

NRI is calculated the number of relation instances extracted by the rules as a fraction of the total possible relation instances in the sentence.

The set of training corpus data and some other data which are not from the corpus are used for the experiment. There are 200 sentences in the collection and reduced typed dependencies of half of the sentences was used to find a minimum value for the Relation Index that should be satisfied by a typed dependencies of a sentence to accept it for corpus update. Other half is used to test the hypothesis. (More sentences can be collected for the experiment to improve the entire process)

Chi Square Goodness of Fit test is used to check the possibility of establishing minimum/maximum value for RI for above mentioned five categories of sentences with respect to extraction rules.

Then the null hypothesis is that RI value can be used to determine the category of the sentence.

From the experiment it was found found that 95% of the category (i) sentences has minimum RI value as 2.0. For category (ii) and (iii) minimum RI value found is 1.0 for 80% of the sentences. 87% of the negative sentences in the category (v) have a maximum RI value as 0.0. RI value of category (iv) sentences lies between 1.0 and 0.00. The table 5.3 summarizes the above mentioned facts.

Sentence Category	RI value	Status
(i)	$2.0 <$	Accept for the corpus update for positive relation sentences
(ii) & (iii)	$1.0 <$	Accept for the corpus update for positive relation sentences with a new extraction rule or new equivalent verb
(iv)	$0.0 < \& 1.0 >$	Reject for the corpus update with respect to the current relation and subject to further analysis for a new relation.
(v)	$0.0 >$	Accept for corpus update for negative sentences.

Table 5.3 The range of RI for each sentence category

Chapter 6

Implementation

6.1 Introduction

GATE[14] the tool used for entity extraction and its extendibility facility used for domain entity extraction is described in Chapters 2 and 4. New extraction rules in Jape which is used for extraction rule construction in GATE's information extraction system ANNIE as mentioned in the Chapter 4 are created for the entities in various domains by incorporating neighbourhood and entity features in to rules.

Section 6.2 describes the common entities provided by GATE's information extraction tool ANNIE for annotation. The individual token types and their attributes and values are described in this section. Two domains, "*Bird*" and "*Sport*" are considered in the experiments for the application of the ontological information extraction techniques developed in the research discussed in this thesis. Reuters-21578 news article corpus is used to perform proposed document classification method based on information extraction technique developed. Section 6.2 further shows methods used in the construction of Jape rules with neighbourhood features for each entity in both domains.

Section 6.3 describes the implementation of the rule based relation extraction. Samples of the positive and negative training data used and the set of rules generated for a relation in the domain *bird* are given here. The initial rule set is thereafter refined to generate the final rule set. The relation instances and the new relations obtained by the application of the final rule set are also shown in this section.

The implementation of document classification with two different types of data is addressed in section 6.4.

6.2 Extending GATE for Domain Entity Extraction

As explained in the chapter 4 GATE's information extraction tool ANNIE [Appendix C] provides a number of common entities which are applicable in many domains. ANNIE **NE Transducer** identifies these entities if they are present in a document, when **Sentence Splitter**, **POS Tagger**, **Gazetteer**, **English Tokeniser** and the **Orthomatcher** are applied on the document corpus. ANNIE **NE Transducer** is capable of annotating

documents with the entities such as *Address, Location, FirstPerson, Person, JobTitle, Title, Organization, Money, Date, Percent* etc. ANNIE embeds extraction rules generated by JAPE to identify the entities. GATE also provides facilities to add the components created by the users to accommodate more domain oriented entities in the system.

The application of JAPE grammar consists of a set of phases, each of which in turn, consists of a set of pattern/action rules. The phases run sequentially and the pattern on the left hand side is matched with the document sentences. When a sentence is matched with the pattern the entity can be identified and annotated by the label specified in the right hand side of the rule. The left hand side of a JAPE grammar aims to match the text span to be annotated, whilst avoiding undesirable matches. For this purpose JAPE grammar requires information from various source components. **Tokeniser** is the main source which splits the entire document into individual tokens such as numbers, punctuation marks and words of different types. A token can have a number of attributes and the attributes and their values are used in the pattern specified in the left hand side of the rule. In addition, JAPE rules can refer to individual word, by the feature "string". Table 6.1 shows token types and attributes in Jape.

Token Type	Attribute	Value
String		String
	Length	Number
Word	Orth	UpperInitial allCaps lowercase mixedCaps
	Length	number
Number	Length	Number
Punctuation	Length	Number
Symbol	Length	Number
spaceToken	Length	Number
Category		Part of Speech tags

Table 6.1 Token types and possible attributes and values [14]

Gazetteer and **Part of Speech taggers** are other source components which are used heavily in JAPE rules. **Gazetteers** are referred to, by the rules in order to extract entities listed as major or minor types (as explained in Chapter 4) in the **Gazetteer**. **Part of Speech Tagger** is used to identify syntactic categories which are required in JAPE rules to specify the token type.

6.2.1 Entity Identification for Selected Domains

JAPE rules normally use gazetteer lookups, characteristics of entities and entity neighbourhood features in the pattern description. For the entity extraction phase of the proposed information extraction methodology a supervised learning approach is used to identify entities. A set of training documents annotated with entities as shown in the Chapter 4 Fig. 4.1 and 4.2 was used in identifying entity neighbourhood features while omitting the determinants. Two domains “*bird*” and “*sport*” are used to demonstrate the generation and application of extraction rules. Table 6.2 and Table 6.3 show the possible entities used in ontology construction from the Wikipedia web documents on the above mentioned domains and the technique used in the development of rules for extraction of those entities.

Entity	Method	Characteristics/Tools			
		Tool	Entity Feature	Neighbourhood Features	
				Pre Neighbor	Post Neighbor
Bird	Gazetteer look up	Gazetteer Bird			
Location	Given by GATE’s ANNIE				
Family	Neighbourhood features		Noun	family	
Diet	Neighbourhood features		Noun	diet, eat, feed on, consists of, food	Eaters

				sources, such as, including	
Weight	Neighbourhood & entity features		Number	weight, weighs/weigh	weigh, g, kg, kilograms. grams, lb
Length	Neighbourhood & entity features		Number	length	long, cm, m, meters, feet, centimeters
Height	Neighbourhood & entity features		Number	height	Tall, cm, m, meters, centimeters, feet
Colour	Gazetteer look up	Gazetteer Color			
Part	Gazetteer look up	Gazetteer Bird_body	Noun		
Eggs	Neighbourhood & entity features		Number	Lay, egg/eggs	egg/eggs
Habitat	Neighbourhood features		Noun	inhabit, habitat	
Nest	Neighbourhood features		Noun	nest in, nesting in	

Table 6.2 Main entities in the domain Bird

JAPE rules are written for each entity in both domains *Bird* and *Sport* and incorporated into GATE framework as plug ins “BirdTag” and “SportTag”. 100 Wikipedia documents have being used in the domain of bird and 75 in the domain of sport as the training corpus [Appendix G]. Wikipedia is chosen as test domains because it gives a good coverage and complicated sentences, being a rich information source for various domains.

Entity	Method	Characteristics/Tools			
		Tool	Entity Feature	Neighbourhood Features	
				Pre Neighbor	Post Neighbor
Sport	Gazetteer look up	Gazetteer Sport			
Location	Given by GATE's ANNIE				
Organization	Given by GATE's ANNIE				
Equipment	Gazetteer look up	Gazetteer Sport_Equipment			
Weight	Neighbourhood & entity features		Number	weight, weighs/weight	weigh, g, kg, kilograms. grams, lb
Length	Neighbourhood & entity features		Number	length	long, cm, m, meters, feet, centimeters
Height	Neighbourhood & entity features		Number	height	high, cm, m, meters, centimeters, feet
Width	Neighbourhood & entity features		Number	width	wide, cm, m, meters, feet, centimeter
Colour	Gazetteer look up	Gazetteer color			
Method	Neighbourhood & entity features		Verb	By	<Equipment>, with, <Equipment>

No_of_Players	Neighbourhood & entity features		Number	played	Players
Game_time	Given by GATE's ANNIE		Number	played	Teams
Play_place	Entity type		Court, Table, Board		
Material	Neighbourhood & entity features		Noun	made out of	

Table 6.3 Main entities in the domain Sport.

6.2.2 Identification Patterns for Selected Entities

The entity and neighbourhood features mentioned in the above tables 6.2 and 6.3 form patterns together with syntactic categories to be used in extraction rules to identify entities. Tables 6.4 and 6.5 show the identification patterns for some entities in the selected domains. Optional features are enclosed in square brackets and entity tags are enclosed in angular brackets.

Entity	Pattern	Example
Family	Family <NN>	family Corvidae
Diet	feed [RB] on <NN> eat <[JJ] NN> diet [of] [BIRD] [RB] consists of <NN> diet [of] [BIRD] VB <NN> food sources such as <NN> food sources including <NN> <NN> eaters	feed mainly on nuts eat small fish diet of Ostrich mainly consists of seeds diet of Parrot is fruits food sources such as nuts food sources including plants, seed eaters
Habitat	inhabit <[JJ][VB] NN> habitat is <[VB] [JJ][VB] NN>	inhabit quiet wooded steppers habitat is open sunny unwooded wetlands
Eggs	lay [up] [to] <number> [color] eggs	lay 2 eggs.

	eggs [] <number> [<number>] [to] <number> eggs	Lay up to eggs normally 4-5 50 to 60 eggs
Nest	nest in [the] [JJ] [RB] [VB] [JJ] <[JJ] Noun> nest on [the] <Noun> nests [VB] [String] [String] <Noun> nest VB [Adverb] built [adjective] in [Determiner] <Noun> nest inside [JJ]<Noun>	nest in colonies. nest in large densely packed noisy colonies. nest on the grounds nests are at times built in nest is usually built high in a conifer nest inside accessible colonies..
Length	[<number> and to] <number m cm meters centimeters cm feet> in height <number m cm meters centimeters cm feet> tall <number m cm meters centimeters cm feet> in length <number m cm meters centimeters cm feet> long	1.8 and 2.7 m in height. 1.1 m tall 76 centimeters in length. 17 cm long
Weight	<number g kg grams kilograms lbs pounds> in weight weigh weighs <number g kg grams kilograms lbs pounds> weigh from <number> to <number g kg grams kilograms lbs pounds> weigh [IN] [<number> [and] <number g kg grams kilograms lbs pounds>	4 kg in weight weighs 22 g weigh from 93 to 130 kg weigh around 25 g

Table 6.4 Identified patterns for some entities in the domain Bird.

Entity	Pattern	Example
Method	<VB> DT [VB] Equipment [with against] [VB] [DT] [Equipment String] <VB IN VB> DT Equipment using [DT] Equipment Equipment <VB VB> IN [DT] [Equipment]	striking a ball with a club trying to maneuver a ball using a hockey stick. Ball is thrown over a net.
No_of_Players	Teams of <number> players	teams of seven players.
Material	made [out] of <NN> <(NN & !Sport)> Equipment constructed of [DT] [NN] [IN] <NN>	made of leather rubber balls constructed of a composite of wood.

Table 6.5 Identified patterns for some entities in the domain Sport

6.3 Design of the Generation of Relation-Extraction-Rules

Having identified entities, the next task is to find the ways that these entities exist in ontology. The GATE output is filtered to keep only the content annotated with the entities retained. This content is sent to Stanford parser to get dependencies which make the base for relation extraction. The dependencies are used to generate extraction rules

From the GATE output, sets of entity instances for each entity are created. Stanford dependencies are required to be filtered out to retain only the relevant clauses according to the criteria explained in the Chapter 5. Attribute values in reduced dependency clauses are placed in the relevant sets of syntactic categories in order to replace those values with syntactic categories or entity names when generalizing the extraction rules. Any value which can be replaced by an entity name will not be considered for syntactic categories.

In the first instance, a java program is used to accomplish these tasks in order to prevent the same procedures being repeated during the rule learning process.. Training set can be updated continuously by the user.

6.3.1 Generation of Relation-Extraction-Rules.

Rule learning is responsible for identifying patterns from dependencies of training sentences to generate extraction rules.

The tasks carried out during rule learning process are

- (i) Generalizing the attributes of all the atomic formulas
- (ii) Ordering the atomic formulas according to the number of occurrences in the dependencies of training sentences.
- (iii) Generating extraction rules
- (iv) Updating the knowledge base.

- (i) Generalizing the attributes of all the atomic formulas

Rule learning process receives the reduced dependencies of the sentences annotated with entities with the attribute values as shown in the Appendix G. Then the dependencies are generalized by replacing attribute values in the atomic formulas with their syntactic constituents or with an entity name if it is an entity instance. Syntactic tagging produced by the *Stanford Parser* is used in this task. Training data set is continuously updated with new additions and the rule learning process should access the training data periodically in order to locate new additions.

Program Design For Generalize

Input: File which contains Dependencies of the training sentences

Output: File which contains Generalized Typed Dependencies

Start

While not end of the file

Get the atomic formula

For both attributes

If the attribute is an Entity

Replace the attribute with the name of the Entity

Else

If the atomic formula is “nsubj” and the attribute is a verb

If the file contains positive data
 Place the attribute in the set of positive verbs for the relation
Else
 Place the attribute in the set of Negative verbs for the relation
End If
End If
 Replace the attribute with name of its syntactic constituent
End If
End While
End

(ii) Ordering the atomic formulas according to the number of occurrences in dependencies

Atomic formulas in dependencies of the both positive and negative training data sets are counted separately and placed in two lists according to the number of occurrences. The atomic formula “*nsubj*” is left out of the lists. This task is achieved by the procedure “*OrderAtoms*” .

Program Design for “OrderAtom”

Input: File which contains the atomic formulas of the positive training examples

Output: List of atomic formulas ordered according to the number of occurrences, Set of positive verbs for the relation

Start

Generalize the attributes of all the atomic formulas

Place all the atomic formulas with generalized attributes in a list

Count the number of elements in the list

Remove all “nsubj” clauses from the list

While the list is not empty

Count all the occurrences of head item in the list

Sort List_Atom according to the descending order of the occurrences of atoms
End while
End

(ii.i) Count all the occurrences of head item in the list

Start
Initialize an occurrence counter to 0
Remove the head from the list
While list is not empty
 If there is an occurrence of the head item in the list
 Increment the occurrence counter
 Delete the occurrence
 End If
End While
Place the removed head item in an another list named List_Atom
Place occurrence counter value in the list Occur_Atom
End

(iii.ii) Sort List_Atom

Start
Count the number of elements in the List_Atom to No_Atoms
For i = 1 to No_Atoms
 For j = No_Atoms - 1 to i+1
 If Occur_Atom[j] > Occur_Atom[j-1]
 Swap Occut_Atom[j] and Occur_Atom[j-1]
 Swap List_Atom[j] and List_atom[j-1]
 End if
 End Loop
End Loop
End

(iii) Generating Extraction Rules

The atomic formula “*nsubj*” is combined with the head of the list of positive atoms in order to form the first rule. The head is then removed from the list and the same procedure is repeated to generate the rest of the rules until all the positive examples are covered by the rules according to ILP algorithm given in Chapter 6. The procedure “ExtractRule” (see Fig. 6.1) embeds the covering algorithm explained in Chapter 6 and implements the task

Program Design for “ExtractRule”

Input: Training Data List_Atom(Sorted list of atomic formulas), File which contains the atomic formulas of negative training data, Set of positive verbs for the relation

Output: Set of Rules for relation extraction

Start

Generalize the attributes of all the atomic formulas of negative examples.

For all the typed dependencies

*Apply the **Covering Algorithm** explained in Chapter 4, Section 4.4.6*

End loop

End

(ii) Update the knowledge base

In the knowledge base there is a set of rules, positive verbs, negative verbs for each relation. This set is updated when new examples to positive and negative training data sets are added by the user and rule learning process itself. The rule learning process is capable of handling some ambiguous situations. . An ambiguous situation arises when

- (a) Relation-extraction-rules cannot identify the existing relation
- (b) Relation-extraction-rules can identify a positive relation, but the verb is not known
- (c) A positive verb is known, but extraction rules cannot extract the relation instances.

On the situation (a), a new relation may be identified and rules can be established for extraction of the relation. In situation (b), the set of positive verbs for the relation will be updated with the new unknown verbs. In situation (c), new rules will be generated for an existing relation. The procedure UpdateKnowledge handles above situations on the received data. Data will be available for the program depending on the situation. If it is the situation (b) a verb constituent will be available. If it is the situation (c) a file containing reduced dependencies will be available.

Program Design for the task “UpdateKnowledge”

Inputs: The file which contains Dependencies of unknown test sentences

Positive verb for a relation

Outputs: New Relation

Updated version of the rule base

Updated set of positive verbs

Start

If the data available is a file

Get the verb constitute from the file

Check whether it is a member of the set of positive verbs.

If the verb is a positive verb

Generate a new extraction rule for the relation

Update the rule base with the new rule

Else

Generate a rule and establish new relation

End If

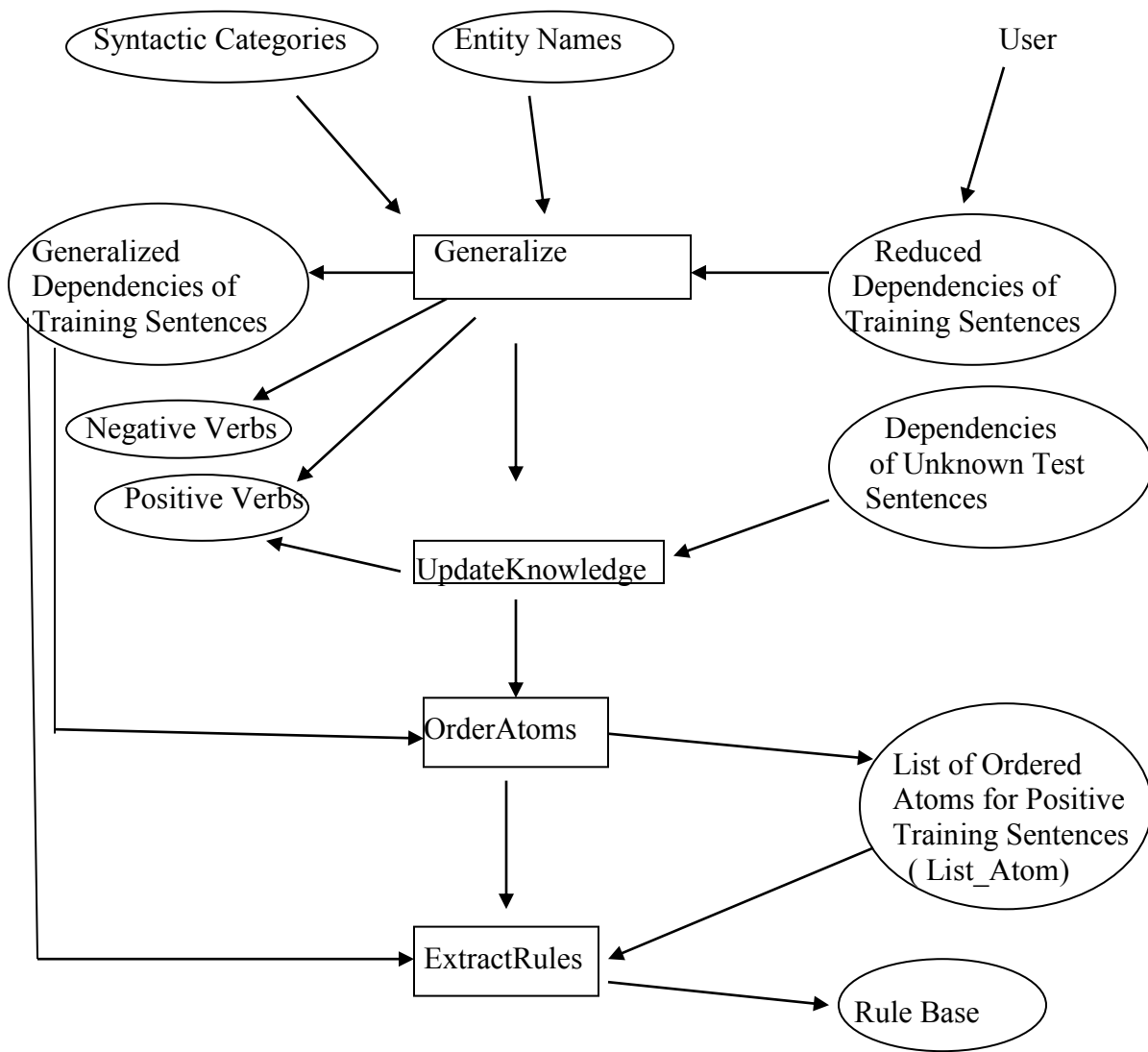
Else

Get the verb constituent for the relation

Add the verb to the set of positive verbs for the corresponding relation

End If

End



6.1 Abstract View of the Rule Learning process.

Tasks are shown in rectangular boxes and data stores are shown in ellipses.

6.3.2 Ontological Relation Extraction

The rules generated on the dependencies of sentences are used to identify relation instances.(see Fig. 6.2)

The tasks performed during relation extraction are

- (i) Extracting relation instances on the application of generated rules
- (ii) Identifying ambiguous situations
- (i) Extracting relation instances

The procedure “*ExtractRelation*” performs Statistical relation extraction by the relation-extraction-rules from the reduced dependencies of test sentences. Identified relation instances are stored as values of the attributes of the relation in order to place them in ontology

Program Design for “*ExtractRelation*”

Input: Set of Extraction rule, Dependencies of Test sentences

Output: Relation Instances, Dependencies of unextracted test sentences with annotated attributes.

Start

For all the Dependencies

Annotate all the attributes with an entity name or its syntactic constituent if the attribute is not an entity

End Loop

While the set of extraction rules is not empty

Apply each rule on the dependencies with annotated attributes.

If a rule covers any of the dependencies

Form a predicate clause with attribute values according to the precedence of the rule.

Calculate the probability in terms of rule weight for the extracted relation instance

If a conditional conjunction is present

Output the condition along with the extracted relation tuple.

If the verb constituent is not in the set of positive verbs for the relation

Store the verb as a new verb for the relation

Delete the dependency of the covered sentence

End If

End Loop

End

(ii) Identifying ambiguous situations

When a situation mentioned in the above 6.3.1.(ii). (a) and (c), is encountered the rule learning process is invoked to pass the information relevant to the situation. The procedure “ReportSituation” pass the information of uncovered sentences to the rule learning process

Program Design for ReportSituation

Input: List of the dependencies of test sentences

Start

If the list of the dependencies of test sentences is not empty

Call the Rule learning process

End

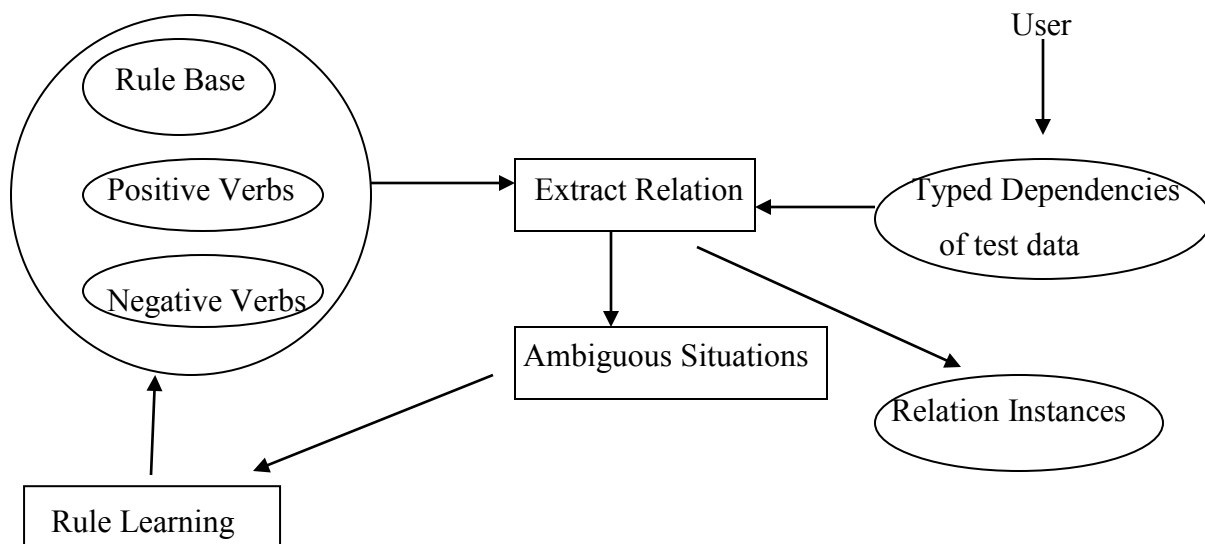


Fig 6.2 Abstract view of the Relation Extraction process

6.4 Document Classification

As mentioned in Chapter 5, entity extraction should be performed on the documents before applying relation-extraction-rules to identify relations present in them. For that purpose documents are passed through the entity extraction phase of all the possible domains or classes in order to identify class specific entities. The threshold for the number of relations that should be existed for the application of classification process needs to be determined.

Program Design for the Document Classification

Input: Weighted relation-extraction-rules,

Threshold and Text document

Output: Class Assignment for the document

Start

Identify entities present in the document

Apply relation-extraction-rules from each class on the document

If the number of relations found in the document from a class is greater than the threshold

Calculate the Class Index CI_{class} as introduced in the Chapter 8

Else

$CI_{class} = 0$

Classify the document with the highest CI_{class}

End

Chapter 7

Experimental Results and Performance Evaluation

7.1 Introduction

In the previous chapters the techniques used in identifying domain ontological concepts and their relations were explained. Applying the same techniques in document classification is also demonstrated with an extensive comparison to other text classification methods. All the extracted concepts may not be related to the domain concerned and all the relationships linking pairs of entities may not be related to a specific relationship. Therefore it is required to verify the accuracy of the extracted information in order to find out whether the extracted information belongs to the domain's scope and they are correctly related. Since the document classification is also based on entity and relation extraction, it totally depends on the accuracy of information extraction.

In the Section 7.2 data sets used in both experiments; information extraction and document classification are discussed along with performance metrics. Entity extraction and relation extraction are evaluated separately by calculating recall, precision and f-measure. Since there aren't any gold standards available for domains experimented, evaluation is mainly based on the test data. But an attempt is made to compare results obtained here with some already published entity/relation extraction results disregarding the domains. However some results of information extraction from the domain 'Sport' can be found in the literature [57] and it is used in the evaluation. In the case of document classification it is possible to do a comparison on the same benchmark data set with a previously published work.

Section 7.3 shows the results of evaluation on ontological entity extraction with comparison to similar systems. Evaluation measures for entities are compared with three other similar systems.

Results of ontological relation extraction with evaluation are shown in section 7.4. However in literature, availability of relevant relation extraction results with proper evaluation to make a comparison with the proposed system was not satisfactory.

Therefore only T. Wang and the team's approach for relation extraction is used in the overall comparison. They have also used features from GATE's language processing components to build a feature vector in their SVM based approach for relation extraction. Carlson's and Yao's approaches are also used to compare the proposed system with respect to two relation types.

Finally in the section 7.5 the results of the evaluation of document classification are shown along with evaluation results of previous work [75]. The previous work [75] published by W. Lam and Y. Han, also includes results of state of art classification methods.

7.2 Data Sets and Performance Matrices

Two different types of data sets are used for information extraction and document classification phases. For information extraction Wikipedia pages from two domains are considered whereas for document classification bench mark corpus on newswire articles which is widely used in classification is used in addition to the domain specific Wikipedia pages which is a good source to cover different sentence structures. These data sets and the way they are adapted in both information extraction and document classification phases are discussed in the following sub sections.

7.2.1 Data Sets for the Experiments

(i) Information Extraction

Two domains are used to test the proposed ontological information extraction methods. Wikipedia pages from domains "*Bird*" and "*Sport*" were selected for the purpose of demonstrating the applicability of proposed method on different domains. Since the training data set is continuously updated by the developed system, rather small number of pages (100) is used at the beginning. Then the rule generation process is continued with the updated corpus to learn new *relation-extraction-rules* which are added to the existing rule base. The pages are used as they are in the Wikipedia for entity extraction. Once the entities are annotated by the system, only language dependencies of the sentences annotated with entities are used to learn the rules for relation extraction. Then reduced dependencies, sets of entities present, the relation verb, sets of other verbs & nouns and

adjectives/adverbs in the sentence are the data sources for rule generation process. Examples of some data samples in both domains for different relations are given in the Appendix (G).

(ii) Document Classification

The domain “*Bird*” is first used to implement the proposed text classification method. Evaluation of relation-extraction-rules on document classification is evaluated in number of ways. First the set of rules were applied on documents annotated with the entities which are embedded in the rules. Secondly it is assumed that the documents are not annotated with main domain entity *Bird*, but annotated with other entities. It is important to do this because a gazetteer of bird names is used to identify instances of the entity Bird. Then it is possible that the gazetteer does not include all the bird names. Therefore when a document is not annotated with entity “*Bird*”, but annotated with many of other entities it still needs to test the applicability of the rules on possible classification of such documents. Finally attempts are made to classify documents in to sub categories in two different ways; five groups according to type’s *peasserine birds*, *wading birds*, *aquatic birds*, *flightless birds* and *sea birds* and three groups according to food that they consume, *carnivore*, *herbivore* and *omnivore*. Extraction rules for seven of relations mentioned in the Table 5.1 in the chapter 5 for the class “*Bird*” were generated. For bird type sub classification one is_a relation was sufficient and for eat type classification two relations were used. The test corpus contains 70 text documents out of which 55 documents are from Wikipedia and 15 are from A-Z animal files. The test corpus contains documents from the classes bird, insects, animals and documents which use the same name as birds; but not from the class *Bird*. For an example two Wikipedia pages which contain bird name Darter; one is from the bird Darter and the other is for fish Darter are included in the test corpus.

A benchmark corpus Reuters -21578 which is widely used in text classification research is also used to test the applicability of the proposed document classification method. The Reuters-21578 collection contains Reuters newswire articles from 1987 in 90 categories. Out of those 90 categories, 7 categories; *acq*, *bop*, *earn*, *jobs*, *dlr*, *trade* and *ship* were selected to be used in the document classification process. At the end the performance of

the proposed method is compared with the results of already published document classification method [75] which also experimented with the Reuters -21578 corpus.

7.2.2 General Evaluation Criteria and Quality Measure

Since entities and relation extraction is performed independently by different methodologies and tools, a different evaluation criterion are used at each stage. Quality measures widely used in information retrieval: Recall, Precision and F-Measure are employed to provide comparable scores of result's quality. In information extraction it is required to check whether the extracted entities are relevant to the domain concerned and relations instances are belong to an identified relation (i.e. positive instance). In document classification the process needs to be verified by examining the assignment of a document to a class.

In information extraction *recall* gives an indication of amount of information extracted. Recall is defined as the number of relevant items extracted divided by number of items actually existed.

$$recall = \frac{\text{number_of_relevant_items_extracted}}{\text{total_number_of_items_existed}}$$

Since the information extraction for ontology construction is a continuous process and domain entities and relations do not represent a finite set the domain scope is restricted to the corpus of documents analyzed in the current study. Then the recall value that is calculated here is the *Local Recall*.

Recall. Local Recall for the entities can be computed as ratio between the number of correctly identified entities and the full set of entities existed in the analyzed corpus.

$$Local_Recall_for_Entities = \frac{\text{Number_of_Entities_Extracted}}{\text{Total_number_of_Entities_in_the_selected_Corpus}}$$

Local Recall for relations (taxonomic or non-taxonomic) is defined as the ratio between the number of relation instances correctly extracted and the number of relation instances existed for a relation in the selected document corpus.

$$Local_Recall_for_Relations = \frac{\text{Number_of_Relation_Instance_Extracted}}{\text{Number_of_Relation_Instances_existed_in_the_Corpus}}$$

Despite its locality this measure can give a good indication of the coverage of the information extraction techniques used on a domain.

Precision is a measure which shows the accuracy of the extracted information. In information extraction *Precision* is defined as the ratio between the number of correctly extracted items and total number of items extracted by the information extraction techniques.

$$Precision_for_Entities = \frac{Number_of_Correctly_Extracted_Entities}{Total_Number_of_Extracted_Entities}$$

$$Precision_for_Relations = \frac{Number_of_Correctly_Identified_Relation_Instances}{Total_Number_of_Identified_Relation_Instances}$$

The *F-measure* can be considered as a verification of a test. It combines Recall and Precision into a single number and accepts a β -value that adjusts the relative importance of recall and precision. Since the precision is more focused here F_β is measured in the evaluation of performances. F-measure scores its best value at $\beta = 1$ and worst value at $\beta = 0$. In information extraction computations β -value is always taken as 1.

$$F_\beta = \frac{(1 + \beta^2) \times Precision \times Recall}{\beta^2 \times Precision + Recall}$$

Local F-measure is obtained by replacing Recall with Local Recall.

In document classification recall and precision are defined as follows

$$Recall = \frac{Number_of_Documents_Classified}{Total_Number_of_Documents_in_the_Test_Corpus}$$

$$Precision = \frac{Number_of_Correctly_Classified_Documents}{Total_Number_of_Classified_Documents}$$

To compare the results published in previous work on document classification two common evaluation matrices are also used, namely the microaveraged recall/precision break-even point measure (MBE) and the macroaveraged recall/precision break-even point measure (ABE). MBE is calculated by averaging the summed up measures for all the classes and taking the break-even point where precision equals recall. ABE is

calculated by taking the average of recall/precision break-even-point of each individual class.

7.3 Evaluation of entity extraction

Evaluation of entity and relation extraction for both above mentioned domains is completely based on the evaluation criteria and measures mentioned in the section 7.2. The entities directly identified by the Gazetteers were not evaluated. The entity type *Location* in both domains is countries. Therefore a Gazetteer is used to identify instances for *Location*. Since entity types *Bird* in the domain *bird* and *Sport, Equipment* in the domain *sport* can be found from a list, Gazetteers are used for those entities.

Results based on the evaluation measures for the entities extracted by JAPE rules in both domains *bird* and *sport* are given in tables 7.1 and 7.2 respectively.

Entity	Precision %	Local Recall %	Local F-Measure
Diet	85	74	79.1
Habitat	87	72	78.8
Nest	87	62	72.4
Eggs	81	89	84.9
Length	84	96	89.6
Weight	92	96	94.0
Family	96	100	97.9

Table 7.1 Evaluation Measures for the basic entities of the domain *bird*.

Entity	Precision %	Local Recall %	Local F-Measure
Length	91.67	100	95.65
Width	91.67	100	95.65
Height	91.67	100	95.65
Weight	91.67	100	95.65
Method	91.67	80	85.44
No_of_Players	100	100	100.00
Material	92.3	73.33	81.73

Table 7.2 Evaluation Measures for some basic entities of the domain *sport*.

Results are compared with some other systems which use similar approach to present work. Since different systems do the evaluations on different domains and give evaluation measures for the entities extracted from domains of their choice, comparison of the performance based on individual entities is not possible. Therefore average measures of all the entities identified by a system were taken in the comparison. A few entity extraction systems which bear a similarity to the proposed system in the context of ultimate results are selected for the comparison. The system Armadillo demonstrates its application on websites of computer science departments to discover the names of people working for a specific department and their personal details including research publications. Therefore results shown below in Table 7.3 for Armadillo are based on the personal details of the researchers employed in computer science departments of various universities and higher educational institutes. Amilcare has shown results based on extracting speaker's name, starting time, ending time and location of a seminar from a seminar announcement corpus. Therefore the entities that they have considered are *Person's name, Time, Location/Venue* etc. Ontoshopie has used short text articles from five different archives in order to demonstrate the extraction of class and event information driven from ontology. Entity is considered as an event and is defined by a concept node with its properties. Ontoshopie's performance had being evaluated for extraction of three entity event classes; *Conference, Award* and *Visiting*. They have conducted four different experiments using three different confidence measures and no confidence measure on the rule set. The best value of four has being chosen here in the comparison.

System	Precision %	Local Recall %	Local F-Measure
Proposed Method	86	81.5	83.67
Armadillo	99.59	85.8	91.76
Amilcare	90	61.5	73.07
Ontoshopie	72.08	15.53	25.55

Table 7.3 Comparison of the presented method with three other systems

Although Armadillo shows the best results out of all 4 systems, the results are completely for the identification of names of researchers who are attached to a particular department. Amilcare extracts entities from semi structured short notices but the proposed method

considers largely unstructured texts which contain long sentences of various degrees of complexity and of various sentence structures. In addition it covers the entities which exist in different ways in the unstructured text. There is room to include new rules to improve the precision; but it make the set of rules unnecessary long with rules which are redundant most of the occasions.

7.4. Relation Extraction

The proposed relation extraction system is used on the domains *bird* and *sport* to extract relation instances existing between annotated entities. Relation-extraction-rules are applied on test documents from Wikipedia and relation instances are identified. Non ambiguous sentences with respect to relation-extraction-rules give the instances for the relations with higher probabilities. Ambiguous sentences as explained in chapter 4 section 4.4.3 contain unknown verbs with rule compliance or known verbs without rule compliance. The relation instances extracted from ambiguous sentences are also given here. The sentences which have neither rule compliance nor positive verbs for the relation may contain ingredients for new relations or give the negative counterpart of the relation. Then the sentences in this category can be further processed to distinguish negative relations from new relations because of the presence of negative clauses in the rules. However all the new relation verbs are considered as negative verbs for the relation concerned in extraction and added to the set of negative verbs.

7.4.1 Extracted Relation Instances and New Relations

Tables 7.4(a), 7.4(b) and 7.4(c) show some results obtained with respect to key relations in the two domains. A set of results relevant to all the relation types shown in table 7.6 are given in Appendix I. For both relations *located_in()* and *related()* new relation verbs are identified by the rules. But these relation verbs are not exactly equivalent to the relation verb of each relation type. Especially in the case of the relation *related()*, the new relation verbs *is_similar_to* and *is_called* are qualified for new relations. Even in the case of relation *located_in()* also the new found relation verbs are more suitable for new relations. Therefore the new relation verbs found by the rules should be further investigated which is done manually at the current stage in order to determine the state of

the new relation verb. There aren't any new relation verbs or new relations found for the relation type *played()* because verbs associated with the entity types “*Method*” and “*Equipment*” are limited. Therefore the equivalent verbs for the relation found during the training phase are sufficient in extracting instances.

Relation Type	Relation Instances found for the Relation	New Relation verbs	New Relations
located_in(Bird, Location)	(Albatross, Southern Ocean) (Petrel, Southern Ocean) (Eagle, Eurasia) (Flamingo, America) (Macaw, Mexico) (Macaw, Caribbean) (Hornbill, Africa) (Hornbill, Asia) (Cassowary, New Guinea) (Kakapo, New Zealand) (Falcon, Europe) (Falcon, North America) (Grebe, South America) (Pelican, France) (Auk, California) (Cuckoo, North America) (Cuckoo, South America) (Cuckoo, Canada) (Eagle, Eurasia) (Eagle, Africa) (Garnet, Southern Africa) (Garnet, Australia) (Garnet, Newzealand) (Spoonbill, Europe)	farmed_in(Bird, Location) (Ostrich, Sweden) (Ostrich, Finland) endangered_in(Bird, Location) (Cassowary, Australia) worshiped_in (Eagle, Peru)	is_national_bird(Bird, Location) (Peacock, India) (Barn swallow, Estonia) (Junglefowl, Sri Lanka)

Table 7.4(a) Results of the relation type *located_in()*

Relation Type	Relation Instances found for the Relation	New Relation Verbs	New Relations
related(Bird, Bird)	(Frigatebird, Pelican) (Falcon, Pelican) (Grebe, Loon) (Grebe, Flamingo) (Shoebill, Hammerkop) (Stork, Herons) (Stork, Spoonbill) (Turcos, Cuckoo) (Swift, Humming bird) (Gannet, Booby)	is_similar_to (Treepie, Magpie) is_called (Kakapo, Owl parrot)	associated_with (Swift, Hummingbird) (Darter, Stork) (Darter, Herons) (Auk, Penguin) prey_for (Duck, Goshawks) (Bat, Barnswallow)

Table 7.4(b) Results of the relation type *related()*

Relation Type	Relation Instances found for the Relation
played(Method, Equipment)	Bandy (direct, ball) (propel, ball) (passing, ball) Discus throw (throw, disc) Pato (throwing, ball) Lacrosse (using, small_rubber_ball) (shooting, ball) Polo (driving, wooden_ball) Tejo (throwing, metal_plate) (throwing, disc)

Table 7.4(c) Results of the relation type *played()*

7.4.2 Evaluation of Relation Extraction

The certainty of the extracted relation instances are measured by the probability calculations done according to the equation 2.5 given in the chapter 2 section 2.5.2. The probability calculations here are based on the dependencies of individual sentences; not on the entire knowledge base which is only used for weight calculations. Each relation-extraction-rule is invoked independently and knowledge base has no impact on the relation extraction. Extracted relation information along with the dependencies can be used to expand the training corpus depending on the probability value and the relation index. Table 7.5 shows the number of instances obtained for the relation *located_in()* according to the rule used in extraction. The column ‘Rule’ refers to the rules in chapter 4 section 4.5. This indicates statistics of the application of rules on the extraction of 75 relation instances.

Rule	Number of Extracted Relation Instances	Probability
1	20	78.76
2	12	62.80
3	23	61.94
4	7	64.56
5	13	84.65

Table 7.5 Statistics of extraction of instances for the relation *located_in()*

The reason for the lowest probability for the rule 3 having the highest coverage is the lower rule weight. The rule 2 is generated because of the frequent presence of *conj_and(location,location)* which also contributes to highest coverage. But rule 2 is the weakest rule and *conj_and* is often present in negative sentences too.

The statistics of the relation types in both domains are given in figures 7.1(a) and 7.1(b). The probability values shown are the values with respect to the best rule in each relation type.

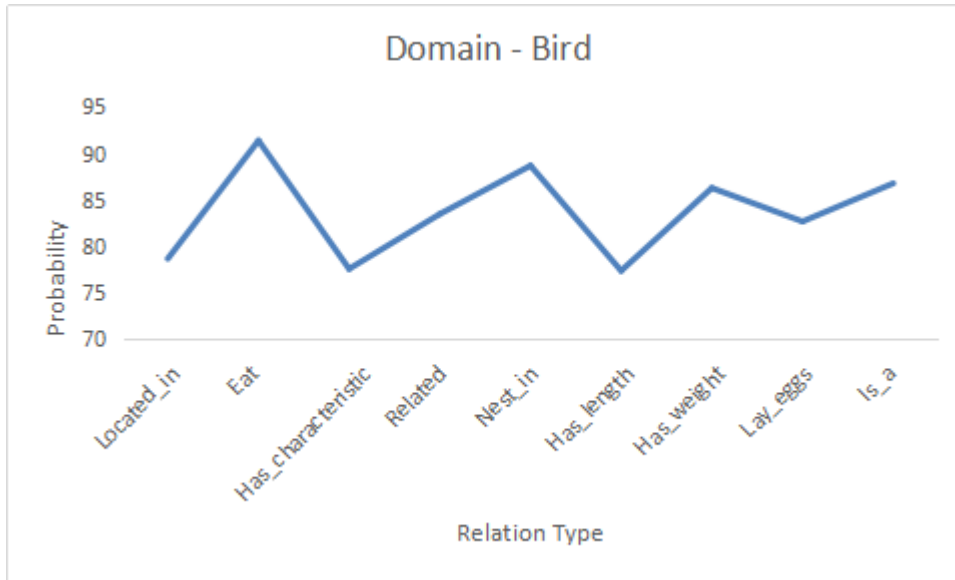


Figure 7.1(a) Probability of Relation Extraction in the domain Bird

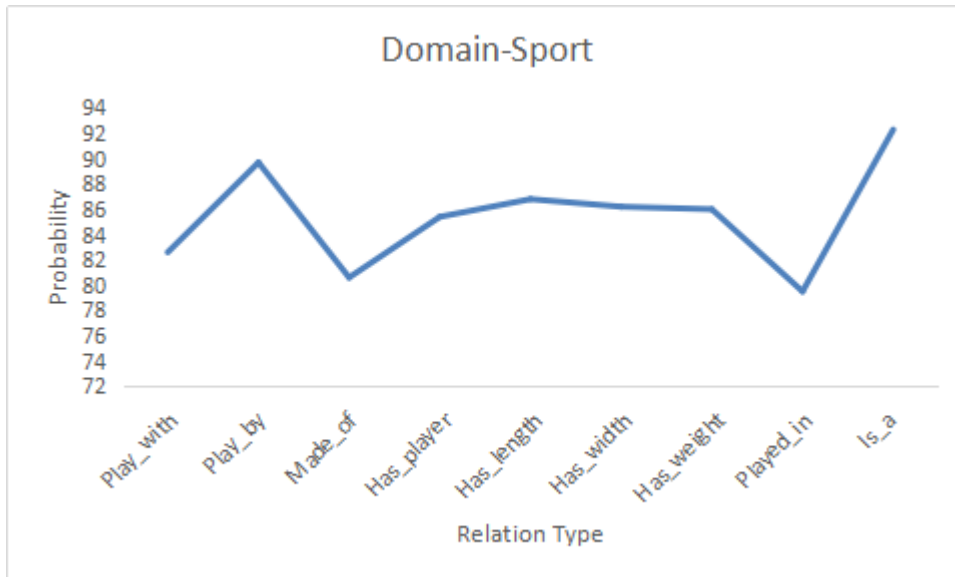


Figure 7.1(b) Probability of Relation Extraction in the domain Sport

Table 7.6 shows the evaluation measures for the relations considered in the domain *bird* and *sport*. Playing method is identified as an entity and “Played” is a common term for all the playing method relations such as striking, shooting, passing etc. Since relation extraction is performed based on the entities identified in the entity extraction phase, both extraction processes are mutually exclusive events. If the entity identification is

inaccurate the relation identification between incorrect entities is bound to be false. However, relation extraction is evaluated independent of the accuracy of entity extraction because techniques have been used and developed for both extraction processes independently. Therefore when evaluating a relation extraction, a 100% accuracy is assumed for extracted entities. Same number of Wikipedia documents have being used in both entity and relation extraction

When analyzing the results few points can be readily identified. In the case of measurement relations *has_length()*, *has_weight()* etc., the main reason for the low precision is the presence of the measurement with comparative adjectives such as more than, less than etc. For an example the sentence “*The ball weights approximately 100 grams more than the volleyball one*” contains the entity types Tool and Weight; but does not give the relation *has_weight()* correctly. Similarly incorrect identification of equivalent verbs for a relation when the verb is unknown obviously has an impact on the precision. In the domain *sport* there are sentences annotated with the entity type *Sport* more than once in many test documents. These sentences can give the relations *is_similar_to(Sport, Sport)* or *is_version_of(Sport, Sport)* which have not been considered in the initial relation extraction task and are important ontological relations.

T.Wang and the team’s approach [48] for relation extraction is selected to compare the performance of relation-extraction-rules. Since the availability of appropriate results for relation extraction similar to relations given here is scarce in the literature, the above mentioned approach is chosen for the comparison despite the fact that it only categorizes relation instances to a number of pre-defined relations. However they have considered a hierarchy of relations with 7 main types and 22 sub types which is comparatively a higher number of predefined relations. Similarly it is possible to compare one individual relation from each domain with two different systems. Evaluation measures used for Relation-extraction-rules in the table 7.7 are averages of all the individual measures.

Domain	Relation	Precision %	Recall %	F-Measure
Bird	located_in(Bird, Location)	83.70	91.21	87.29
	eat(Bird,Diet)	88.22	68.01	76.81
	has_characteristic(Bird, Bird_Part)	80.26	83.67	81.93
	has_characteristic(Bird_Part, Feature)	80.82	76.43	78.56
	related(Bird, Bird)	72.48	89.30	77.15
	nest_in(Bird, Nest)	77.52	65.55	71.03
	has_length(Bird, length)	70.56	91.60	79.72
	has_weight(Bird, weight)	71.80	90.48	80.06
	lay_eggs(Bird, Egg_number)	91.38	90.65	91.01
	is_a(Bird, Super_bird)	92.01	89.29	90.63
Sport	play_with(Sport, Equipment)	76.75	86.74	81.44
	played(Method, Equipment)	62.15	74.06	67.58
	made_of(Equipment, Material)	80.01	82.17	81.08
	has_player(Sport, Player_number)	67.50	72.40	69.86
	has_length(Tool, Length)	72.07	80.25	75.94
	has_width(Tool, Width)	73.98	80.01	76.88
	has_weight(Tool, Weight)	68.25	83.88	75.26
	played_in(Sport, Location)	84.08	92.50	88.09
	is_a(Sport, Super_sport)	92.63	94.06	93.34

Table 7.6 Evaluation Measures for relations from two domains

System	Precision %	Recall %	F-Measure
Relation-extraction-rules	79.95	92.6	85.8
T.Wang's team Approach	73.87	69.5	71.59

Table 7.7 Comparison of performance of relation-extraction-rules with Wang and the team's system

In the domain Bird the relation *located_in()* can be compared with the relation *liveIn* from Yao at el's approach [61] though arguments of *located_in()* relation are *Bird* and *Location* whereas arguments of the *liveIn* relation are *Person* and *Location*. In the domain Sport the relation *play_with()* can be compared with the relation *SportUsesSportsEquipment* from Carlson at el's approach [57]. It uses constraints to couple semi supervised learning as explained under related work. Three coupling algorithms CPL, CSEAL and MBL have being developed in their approach to information extraction as explained under related work. Results are shown in table 7.8.

Although the CSEAL algorithm achieves 100% precision it is claimed in the publication that the MBL gives the overall best performance and CSEAL incurs some loss in recall.

Relation	Yao at el	Carlson at el			Relation-extraction-rules
		CPL	CSEAL	MBL	
located_in()	56				83.7
play_with()		33	100	33	76.75

Table 7.8 Comparison of the Precision with two different approaches for two different relations.

Although a smaller number of training examples are used to initiate the system, it will not affect the performance of the system because any situation that cannot be covered by the extraction rules is considered as an instance for a new relation and a new extraction rule is generated for the relation accordingly. In addition to that the training set is continuously expanded by the information relevant to extracted relation instances, bearing a resemblance to semi supervised or bootstrapping method. Therefore use of a smaller training data set becomes an advantage here and has no adverse effect on the performance of the entire system. With the expanded training corpus the performance of the system is expected to be improved further.

7.5 Evaluation of Document Classification

Out of the two domains used for the experiments under the relation extraction the domain *bird* is used in document classification. The benchmark corpus Reuter-21578 which is widely employed for text classification based researches is also used here to demonstrate the applicability of the proposed method on two different types of text corpora.

The results of performance metrics of document classification are presented with respect to the parameters; the minimum number of rules applicable for correct classification N which is given as a percentage of the total number of rules in the system and the Class Index CI .

7.5.1 Evaluation based on the Selected Domain

All performance measures were calculated based on test corpus. Therefore the performance measures shown in the Table 7.9 and 7.10 are some local recall and

precision values along with two f-measures on the text corpus in the domain *bird*. Figure 7.2 gives the graphical representation of full set of local recall/precision measures in this domain calculated on the variations of *N* and *CI*.

Classification	Recall %	Precision%	F(1)	F(0.5)	N(min)%	CI(min)
Fully annotated documents	61		76	88	57	0.5
	84		91	96	42	0.4
	90		94	97	28	0.3
	97		96	96	28	0.2
Partially annotated documents	68	91	77	85	57	0.5
	88	83	85	83	42	0.4
	95	75	83	78	28	0.3

Table 7.9 Classification performance with respect to *N* and *CI* in the domain *bird*

Class	Recall %	Precision %	F(1)	F(0.5)
Pesserine	94	100	96	98
Wading	100	100	100	100
Flightless	96	100	97	99
Seabird	100	100	100	100
Aquatic	94	100	96	98
Overall (eat types)	82	89	85	87
Carnivore	77	95	85	91
Herbivore	82	67	74	70
Omnivore	82	90	86	88

Table 7.10 Sub classification of the main class Bird on Bird type and Eat type

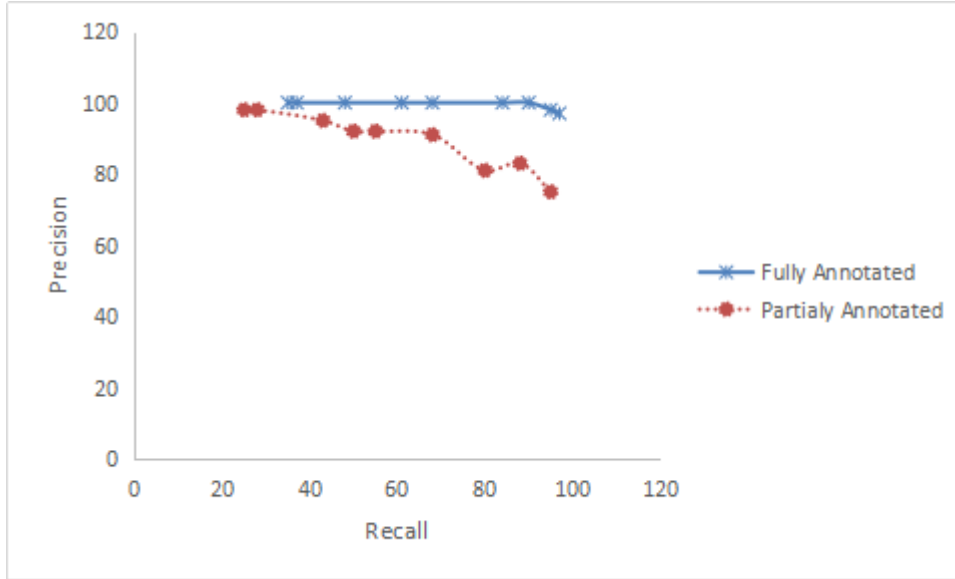


Figure 7.2 Recall/Precision performances of fully and partially annotated documents

7.5.2 Evaluation based on a Benchmark Corpus widely used for Text

Classification

Only seven categories from Reuters-21578 news wire corpus is selected for the classification task. Since proper sentences are needed to generate relation-extraction-rules short articles with tabular data were not considered. Categories selected with relevant entities and relations are shown in the table 7.11(a), (b), (c), (d), (e), (f) and (g). Linguistic patterns for the entities in the corpus is given in Appendix J. Table 7.12 shows local recall and precision values with two f-measures on the selected 7 categories of Reuters-21578 news wire corpus. The variation of recall and precision based on our parameters are shown in the Figure 7.3 and the comparison with other text classification methods is given in the Table 7.13. The MBE and ABE values for the all the methods are directly taken from previous work [75]. According to them their values are based on all 90 categories of the news-wire corpus whereas our values are based on only 7 categories. But the comparison is done to demonstrate the applicability of the proposed method on a benchmark corpus

Category	Entities	Relations
Acq	Company/Organization, Share_price, No_of_shares, Product, Profit, Purchase_price, Income Service	acquire(Company, no_of_shares) acquire(Company, share_price), sell(Company, no_of_shares), sell(Company, share_price), Sell(Company, Product/Service), Sell_to(Company, Company) Merge_with(Company, Company), has_profit(Company, Profit), earn(Company, Income) Provide(Company, Product/Service)

Table 7.11(a) Category *acq*

Category	Entities	Relations
Dlr	Currency, Currency_Rate Country, Dollar_Rate, Period, Dollar_Rate_Surplus, Rise_in_Export, Rise_in_Import	has_dollarValue(Currency, Dollar_Rate) has_dollarValue_in(Dollar_Rate, Year) has_value(Currency, Currency_value) rise_import_with(Country, Rise_in_Import) Rise_currency_rate(Currency, Currency_Rate) rise_currency_rate(Currency, Currency_rate) post_dollar_rate_surplus(Country, Dollar_Rate_Surplus)

Table 7.11(b) Category *dlr*

Category	Entities	Relations
Bop	Current_account_surplus, Current_account_deficit trade_surplus, Export, Import, Inflation_rate. Burrowing, Country Company	has_deficit(Period, Current_account_deficit) has_deficit(Country, Current_account_deficit) has_account_surplus(Country, Current_account_surplus) has_trade_surplus(Period, trade_surplus) has_trade_surplus(Country, trade_surplus) import(Period, Import) export(Period, Exports) has_burrowing(Period, Burrowing)

Table 7.11(c) Category *bop*

Category	Entities	Relations
Trade	Inflation_rate, Debt, Trade_surplus, Period, Export, Import, Country,	has_inflation_rate(Country, Inflation_rate) has_debt(Country, Debt) has_trade_surplus(Country, Trade_surplus) has_surplus_in(Country, Period) increase_surplus(Trade_surplus, Trade_surplus) rise_exports_to(Export, Export) fall_imports_from(Import, Import)

Table 7.11(d) Category *trade*

Category	Entities	Relations
Earn	Stock_amt, Company, Period, Profit_amount, Expense, Sales_price, Income, Service_Unit, Dividend, Sales, Product, Product_value,	increase_stock(Stock_amt/Company, Stock_amt) has_stock(Company, Capital_stock) has_profit_below(Company/Period, Profit_amount) has_expenditure(Company, Expense) has_profit(Company, Profit_amount) has_sale(Company, sales) earn(Company, Income) has_sale(Company, Service_Unit) declare_dividend(Company, Dividend) pay_dividend(Company, Period) produce(Company, Product)

Table 7.11(e) Category *earn*

Category	Entities	Relations
Ships	Shippingline, Port, Period, Capacity, Ship, Good, No_ships, No_people, Location, Weather, Country,	transfer_charters_to(Shippingline, Shippingline) has_capacity(Port, Capacity) halt_at(Ship, Port) is_closed(Port) carry(Ship, good) lease_ships(Shippingline, Shippingline) lease_ships_for(Shippingline, Period) has_agreement(Shippingline, Country) carry(Ship, No_people) rescue_from(Ship, No_people) sale_from(Ship, Port) is_halt(No_ships, Location)

Table 7.11(f) Category *ships*

Category	Entities	Relations
Jobs	Country, Period, Rate, Industry, Company, Employment	has_unemployment(Country/Period, Rate) has_femaleUnemployment(Country/Period, Rate) has_maleUnemployment(Country/Period, Rate) had_employment(Industry, Rate) rise_employment(Employment, Rate)

Table 7.11(g) Category *jobs*

Entities and relations which appear in less than 5 documents were omitted from using in the classification task. In the category “ship” most of the news articles are based on ship disasters, poor weather conditions and trade union actions in shipping organizations. But news articles based on trade union actions were omitted from the training corpus in order to reduce the number of rules being idled most of the time during the classification process.

Class	Recall(%)	Precision(%)	F(1)	F(0.5)
Acq	82	70	76	72
bop	78	62	69	65
dlr	85	78	81	79
earn	67	78	72	76
jobs	80	75	77	76
ship	75	80	77	78
trade	67	85	75	81

Table 7.12 Classification performance of the selected categories of Reuters -21578 document corpus

	Rocchio	WH	KNN	NN	SVM	GIS-R	GIS-W	Relation-extraction-rules
MBE %	77.7	82	80.2	80.7	84.1	83.0	84.5	75.2
ABE %	57.8	64.9	60.7	59	64	62.5	65.5	70.6

Table 7.13 Comparison of the proposed method with the results of state of art text classification methods and improved methods on Reuters -21578 document corpus

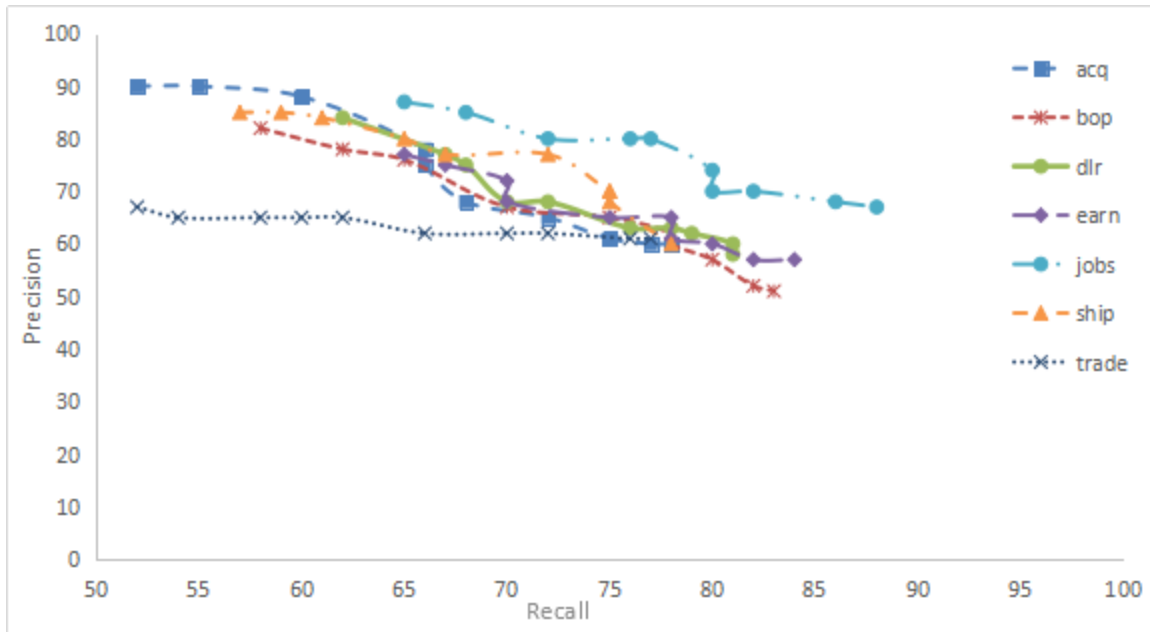


Figure 7.3 Recall/Precision performances of 7 categories in the Reuters -21578 corpus

The remarkable point in results is the 100% precision in most of the occasions in the domain *Bird*. Since relation-extraction-rules contain class specific terms (class specific entities) there is little room for misclassification. In addition the entities appeared in the documents are very less in number when compared to different words appeared in a document. Therefore misclassification due to irrelevant terms is prevented to a great extent. Especially the *relation-extraction-rules* are applied on the sentences annotated with two entities except the rule for the relation *is_a* which contains one entity and the class name. A significant drop in the precision is shown in the case of partly annotated documents where majority of the annotated sentences are annotated with only one entity type because the main entity type is left out. Then it is expected to use the rules to identify the missing entity by correct classification of the documents. In sub classification task the reason for the highest precision is that a simple sentence is available in the documents which contain above mentioned sub category information, to capture the taxonomic relation *is_a*. Less number of documents available in the category *herbivore* in the eat-type sub classification is a possible reason for the low precision in the category. A rather small set of *relation-extraction-rules* is used with an average of three rules for each relation in the domain *Bird*. Therefore the rule set might not have been sufficient to capture all seven relations used in a document resulting in a poor recall with higher

threshold for the number of rules. On the other hand some documents do not contain sufficient information to identify most of the relations covered by the rule set. That is the reason for increased recall values when the threshold for number of rules is reduced. Our results shows that best recall and precision can be achieved even with low N value as 28% which is the case with two relations out of seven relations and low CI value in this domain.

In Reuters-21578 corpus the lengths of documents are much less than Wikipedia pages. Some relations are overlapping in the categories *earn*, *bop*, *acq* and *trade*. Therefore overall classification performances are low in those categories. In the category trade most of the articles contains details of the individual incidents related to trade sector. Each incident is different from the other. Therefore only few relations out of all relations applicable to the category are present in one document resulting low precision values in the classification. Although proposed method has shown a comparatively low MBE value, results indicate the applicability of the method on different domains with various lengths of documents with good performance measures. Use of relation-extraction-rules is more effective on rather large documents which provide considerable number of relations.

Chapter 8

Conclusions and future work

8.1 Summary

Inspired by the enormous amount of information on the World Wide Web this research work was strongly motivated for the formal representation of information available in various repositories in order to make information retrieval more efficient and meaningful. However, the unstructured, unreliable, duplicated nature of the web documents have made formalizing information through a representation such as an ontology, a complicated task which needs the attention of information technology researchers. Therefore an attempt has been made here to address the issue by paying attention to the most challenging task; extraction of information from various information sources for dynamic ontology construction/population. In this thesis emphasis is placed on basic aspects of ontology (i.e. Concepts represented by Entities, Relations and Instances).

In developing methodologies for information extraction a higher weight is given to non-taxonomical relations which are yet to be addressed widely and efficiently. With regards to ontological entities the focus is for extracting entity instances for known entity classes. Domain entity classes can be taken from already existing reference ontology or manually figured out by a domain expert. Since the entire system is based on supervised learning, training data is used to identify entity neighborhoods as patterns of lexical terms and their types in order to generate extraction rules. Relation extraction is handled separately by generating relation-extraction-rules. Further the use of relation-extraction-rules is extended to document classification to demonstrate its applicability on a related area. Finally the evaluation is done according to a general evaluation criterion and a comparison is made with few other systems having similar objectives.

8.2 Evaluating Research Questions with Concluding Remarks

A number of research questions with respect to ontological information extraction have been identified and investigated. To recap,

- Processing natural language sentences towards identification of relations: which characteristics of natural language sentences can be used and how they can be effectively used in relation extraction?
- Generation of rules for extractions of relations embedded in natural language sentences: How can the natural language characteristics be used in the extraction rules? How can the rules be generated to achieve a high precision by avoiding extraction of false relation instances?
- Assigning weights for the relation-extraction-rules: How can weights be properly assigned to measure the strength of the extraction rules?
- Testing the applicability of relation-extraction-rules on a classification task as proof of concept: Is it possible to address the problems encountered with the state of art classification methods by using ontological information? How can the relations-extraction-rules be used in classification for improved performance?
- Using dynamic training corpus on supervised learning: How the training corpus can be expanded automatically with the extracted information to make the time and manual labour consumed in creating a large training corpus avoidable?

8.2.1 Processing natural language sentences towards identification of relations

In extracting information from domain relevant documents natural language sentences should be analyzed to identify domain specific terms which can be categorized into ontological entity classes when the term is a noun. When the term is a verb it may belong to a domain specific relation type. Effective information extraction from natural language text requires preparation of the text for the task. Therefore languages parsing being an

essential initial step in natural language processing, language researchers have already made their contributions with a number of well-established parsers. Stanford parser is selected mainly because its output contains dependencies of lexical terms of a sentence in addition to parse tree with syntactic tags. In relation extraction these dependencies are searched thorough in order to identify the main verb constituent and the lexical pattern surrounding it. A simple heuristic is used to determine the relation verb when there is more than one verb in the sentence. As such the dependencies can be directly investigated with part of speech tags and entity types in the sentences which are annotated with entity types found during entity extraction phase. In the chapter 4 section 4.3.1 a criterion is introduced in order to filter out the unnecessary dependencies and combine the lexical terms wherever appropriate. Therefore dependencies are reduced to clauses with lexical terms around the entity types. Further the conditional conjunctions can be identified and then extracted relation instances can be given with a condition under which the relation exists.

Both taxonomic and non-taxonomic relations are treated in the same way as dependencies provide a very reliable source for finding predicate clauses specific to different relations. Any sentence can be accommodated in the system irrespective of the complexity of the sentence, paving way to non-taxonomical relation extraction with known or unknown relation types.

Dependencies can also be effectively used for entity extraction. Same method could have employed for entity extraction though GATE is used for the purpose because it already provides resources to extract commonly used entities and can be extended for domain entities. Patterns similar to entity neighbourhood patterns can be searched from dependency clauses to form extraction rules. As a result the number of false entities can be reduced because complete sentences and not just text phrases are considered in generating extraction rules.

8.2.2 Generation of rules for extractions of relations embedded in natural language sentences:

Chapter 4.4 discusses the use of linguistic characteristics combined with the Inductive Logic Programming Technique to learn rules for relation extraction. ILP plays the major

role as the learning algorithm to induce relation-extraction-rules. ILP used in the learning mechanism prevents the extraction of negative relations. The heuristics and ILP algorithm used, gives the best possible set of rules generated based on the training corpus at the beginning. Therefore additional refining techniques are not necessary to select best rules. But the initial rules are augmented with dependency clauses relevant to negative sentences for the relation in order to avoid extraction of negative relation instances. The use of ILP technique to generate rules automatically from the dependencies of natural language prevents the system from generating arbitrarily complex rule sets which would be difficult to understand or maintain. Natural language texts suffer from semantic ambiguity. Success of information extraction from natural language text is affected by this aspect. Since linguistic resources such as wordnet only provides synonyms for a word and synonyms only are not sufficient, the system maintain its own resources to keep equivalent verbs (both positive and negative) within the purview of the relation in order to reduce the effects of semantic ambiguity. Then the rules are again augmented with negation of negative verbs to avoid extraction of negative relation instances. The set of positive verbs is only looked up in an ambiguous situation. Another positive aspect of the approach is that the ability to identify relations which cannot be categorized into pre-defined relations. Therefore there are no relations of unknown category between known entities. Use of a rather small text corpus as training data can be justified by provisions made for continuous expansion of the text corpus and by the selection of the text corpus which covers a number of different syntactic structures. The biasness of the data set, it being selected only from Wikipedia does not have an adverse impact on the final outcome as it is a good hierarchical information source for many domains and it is conveniently readable by the general public.

8.2.3 Assigning weights for relation-extraction-rules:

Once the relation-extraction-rules are generated proper measures should be devised to assess the strength of a rule. Log odd ratio weights or priority based confidence measures only give an indication over the coverage of the rule on the training data set. In information extraction by rule based systems, the validity of the extracted information depends on the strength of the rule used in extraction. Therefore the weight of the rule

should reflect the accuracy of extracted relation instances. This is explained for the example relation *located_in()* with the rules generated for the relation in chapter 4 section 4.5. In chapter 7 under the evaluation, probability of the relation instances extracted by each relation-extraction-rule for the relation *located_in()* is given by the table 7.5.

Markov Logic Network (MLN) provides a suitable environment for relation-extraction-rules. Modeling relation-extraction-rules in MLN environment for statistical relation extraction makes way to find weights for the rules. Weight of a rule has a direct effect on the probability calculation of the relation extraction in MLN. Therefore the extracted relation instances can be produced with a probability value according to the rule employed in the extraction as in table 7.5. Even in the absence of an equivalent verb for a relation in an annotated sentence, a relation instance can be extracted with a probability as long as dependencies of the sentence can be covered by a relation-extraction-rule. Later the set of equivalent verbs for the relation can be updated with the new found verb depending on the probability and the Relation Index (RI) value of the sentence.

Computations in MLN are intractable because of the large number of atomic formulas available after grounding the formulas with constants from the training data. Therefore the initial rule set only with three atomic formulas before augmenting with negative clauses is taken for weight learning process. It is assumed that there won't be a significant difference in the final weights even though the complete rules were considered. Justification of using only the initial version of rules is given in chapter 4, section 4.5 under 'weight learning'.

8.2.4 Testing the applicability of relation-extraction-rules on a classification task as proof of concept

Feasibility of the information extraction method developed is tested on document classification task. Since the rule based information extraction system is constructed on training text corpus which is gathered from a domain specific text resources, collecting documents in a specific domain is the very first important task of the whole process. Then a domain can be considered as a class and the set of entities and relation-extraction-rules are the signature for the class. Sub classification can be done within the domain by rules appropriate to sub classes. It is investigated whether the relation-extraction-rules along

with entities can be used effectively on document classification for improved performance based on two parameters; Classification Index (CI) and the number of rules applicable.

Selecting the set of class specific features is the major challenge in traditional text classification methods which can end up with large number of irrelevant features. Rule based classification systems can contain large number of lengthy rules with many conditions. But replacing the class specific features by entities and relations as in the proposed information extraction based method, contributes to reduction in number of features. In addition to that it takes account of the semantic of the class specific features in the form of domain specific entities and their relations which leave the irrelevant terms out. Classification by semantic features is bound to be more accurate than classification by the presence of selected features.

Document classification by ontological information is experimented with two types of test data. Test data from the domain *Bird* and a benchmark corpus for text classification *Reuters-21578* are used to demonstrate the applicability of method on general text classification tasks as well as on domain based classification tasks which include filtering out documents for a specific domain and classification within the domain. The use of ontological information is tested on the domain *Bird* to separate documents belong to the domain from a collection of documents in different domains. Same technique is used to do a sub classification and a simple hierarchical classification. Seven categories from the *Reuters-21578* are selected to test the applicability of the method. Then classification results shown in chapter 7 section 7.5 indicates that this information extraction method is more effective in domain based classification though it can be applied satisfactorily on the *Reuters-21578* too. The main reason for this may be the number of rules applicable in a document. Since the selected Wikipedia pages are long, number of relation-extraction-rules applicable on a document is comparatively higher than number of rules applicable on short news articles from Reuters-21578 corpus. Besides some news articles report different events in the same category. For an e.g. in the category “ship” there are three types of articles which gives three different specific type of entities and relations in addition to few common entities and relations, leading to a clear separation boundaries

for three sub categories of the main category. Therefore few event specific entities and relation-extraction-rules are applicable to the short articles in this category.

When relation-extraction-rules are used for document classification in two types of test data it is shown that this information extraction method is well effective in hierarchical classification and sub classification. On some occasions only one or two rules are sufficient in sub/hierarchical classification. Although the word based classification methods can also applied with less number of features in sub classification these specific features along may not give a clear indication for separation with lengthy documents. But with relation-exaction-rules the document should be searched only for few relations relevant to a sub category.

8.2.5 Using dynamic training corpus on supervised learning

Although the supervised learning is less complex and can achieve higher accuracy, the time and labour consumed in creating a large training text corpus is not compensated by it. Especially in domain ontology construction compiling a large training text corpus initially for different domains is not feasible. Therefore initiating the process with a small training corpus and expanding it during the process will give the performance of a supervised learning but avoiding time consuming laborious task of creating a large training corpus. When this corpus expansion process is accomplished extraction rule induction will be based on a dynamic corpus. Hence there are rooms for the rule base to grow for improved performance. The proposed system has the provision to expand the training corpus with both positive and negative sentences for relations. The sentences which are neither compatible with extraction rules nor relation verbs are assumed to provide new relation types. Negative sentences for relations can also be identified by the presence of negative verbs or negative dependency clauses. But there are no provisions to identify unknown negative relation verbs. Non-ambiguous sentences as well as ambiguous sentences can be used to expand the training text corpus. Sentences which give acceptable relation instances are added to the corpus in order to expand it and improve the performance.

At present a simple measure Relation Index (RI) is proposed to use in expanding the training corpus with the information of extracted relation instances and updating the set

of relation verbs with new found equivalents. RI value exploits sentence characteristics unused in the generation of relation-extraction-rules. RI value is combined with a probability threshold in order to accomplish the automatic expansion of the text corpus. But automating the corpus expansion can be addressed extensively with more sophisticated techniques which are discussed under the future work in section 8.3. Researching extensively on the training corpus expansion is out of the scope of the thesis.

8.2.6 Overall Concluding Remarks

As the extensive language processing tool GATE provides a very good environment for entity instance extraction along with a number of necessary language processing facilities it is chosen for domain entity extraction. Neighbourhood pattern matching fits well in the jape rules which are generated from training data set and GATE can simply be updated with new sets of rules as processing resources for different domains. Therefore as long as the memory requirement is fulfilled, new processing resources can be plugged into GATE to be used for entity extraction in various domains. Since GATE provides the basic general entities which can be come across in many domains, only domain specific entities are required to be considered in generating extraction rules. Therefore GATE is used comfortably and confidently for domain entity extraction purposes.

The system is more suitable for domains from which ontology can easily be pre modeled with entities, as the work is initiated based on manually crafted ontology model or list of possible entities. The objective was not to find new entities, but to extract entity instances for already identified entity classes. Highly specific domains might not be very good candidates for the system. Even in the demonstration of the system two domains which have drawn a wide range of users irrespective of age, nationality, country and social differences etc. are selected. But the system is still applicable in highly specified domains when entity types are identified by some other means, either by a simple method as finding the frequencies of occurrence of domain specific words or by a more sophisticated method taken from the literature. When a simple method is used as mentioned above, the list of entities obtained can be refined with manual intervention. Therefore the system can be adapted to any type of domain.

An added advantage of the system is that the same set of rules can be tried in different domains as well as the same techniques are applicable to different domains. Then the entity types will be replaced with the entities specific to a domain if the rules comply with any of the annotated sentences.

In overall this research has managed to exploit available tools and techniques for entity extraction and linguistic characterization to be used by ILP, for successful relation extraction. Applicability of information extraction on document classification is demonstrated by two types of data sources. Learning process can be considered as a version of semi-supervised learning. In semi-supervised learning large amount of unlabeled data is used with a small amount of labeled data. In the proposed approach test data is the unlabeled data and extraction rules induced from labeled training data are applied on test data to find relevant information. The difference here is the extracted information is used to expand the training data set and the process can be repeated with expanded corpus until no significant further improvement is incurred.

Although the rule based systems are being replaced by statistical machine learning in the academic researches, rule based information extraction dominates the commercial world [84]. Therefore hybridized systems have a significant impact on the field of information science and extraction. The rule based information extraction system described in the thesis hybridize with statistical machine learning in automatic rule induction and weight learning for the induced rules in statistical relation extraction.

8.3 Future Work

In this section several future directions of the research are described

- In entity extraction only neighbourhood patterns are used because entity and relation extraction are treated separately and relation extraction is the main focus of research. In addition, entity or domain concept identification is being widely addressed and many techniques have been already developed for this purpose. Therefore these techniques can be used in a new module as a processing step and can be incorporated into GATE in order to enhance the current process. Noun phrase based patterns [43] is such a strategy that can be used as a new plug_in to improve performances of the system. The added advantage is that the recall of taxonomic relations can be improved by annotating the

sentences with taxonomical entities(i.e. entity class, a sub class of the entity class .and entity instance)

- Extraction rules can be generated for entity extraction too as for relation extraction mentioned in chapter 4. As a result the accuracy of the entity extraction can be improved by verifying the already extracted entities by the application of new extraction rules generated from a different method.
- Since any type of sentence can be considered irrespective of the complexity of the sentence, identifying the correct dependency between two entity instances itself can be a complicated task. At the present state the dependencies are reduced strategically and employ a simple heuristic to address the issue is employed. However the accuracy of the process can be improved when attention is paid to analyze long complicated structured sentences to make them simpler by restructuring or by breaking the sentence at an appropriate point if necessary.
- In the present method sentences with unknown verbs and cannot be covered by the rules are rejected for the corpus. Training corpus expansion can be researched further in order to enhance the use of dynamic corpus. One suggestion is to use a fussy neural network with an appropriate representation for a sentence whose candidature for corpus expansion is determined by the output of the neural network. Relation verbs can be categorized mainly with respect to the preposition used with the verb and predefined basic word sequences which contain the two entities and the verb can then be established for each verb category. Then Relation Index (RI), n gram probability and deviation from pre-defined word sequence can be used as input values for the network, representing sentences. Negativity of a sentence for a relation can also be incorporated with these values in sentence representation or negativity can be handled separately. External resources such as wordnet, freebase etc. can be incorporated to aid the identification of verb when the verb is unknown.

Bibliography

1. Aitken J.S., (2002) Learning information extraction rules: an inductive logic programming approach, Proceedings of 15th European Conference on Artificial Intelligence, pp 355-359, Lyon, France
2. Baungartner, R., Enzi, C., Henze, N., Herrlich M., Herzog M, Kriesell M., Tomaschenski K., (2005) Semantic Web Enabled Information Systems Personalized Views on Web Data., Proceedings of International Ubiquitous Web Systems and Intelligence Workshop., Suntec Singapore.
3. Buitelaar P., Olejnik D. and Sintek M.(2003) OntoLT: A protégé plug-in for ontology extraction from text., Proceedings of the International Semantic Web Conference.
4. Berners-Lee T., Hendler J. and Lassila O (2001) The Semantic Web., Scientific American, May 2001.
5. Celjaska D., Vargas-Vera M. (2004) Ontosophie A semi-automatic system for ontology population from text, International Conference on Natural Language Processing.
6. Chandrasekaran B., Josepson J.R., Benjamins V.R. (1999) What are Ontologies and why do we need them, Intelligent Systems and Their Applications, IEEE 14-1 pg. 20-26.
7. Cimiano P., Volker J. (2005) Text2onto – a framework for ontology learning and data driven change discovery, Int. Conf. on Applications of Natural Language to Information Systems.
8. Ciravenga F. and Wills Y.(2003) Designing Adaptive Information Extraction for the Semantic Web in Amilcare, Annotation for the Semantic Web, in the Series Frontiers in Artificial Intelligence and Applications by IOS Press, Amsterdam.
9. Ciravenga F., (2001) (LP)², An Adaptive Algorithm for Information Extraction from web-related texts, Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management.
10. Ciravenga F., Chapman S., Dingili A., Wilks Y. (2004) Learning to Harvest Information for the Semantic Web, Proceedings of the 1st European Semantic Web Symposium, Greece.
11. Craven M., DiPasquo D., Freitag D., McCallum A. K., Mitchell T.M., Nigam K., Slattery S. (2000) Learning to construct knowledge bases from the World Wide Web. Artificial Intelligence 118(1/2) 69-113.

12. Craig A. Knoblock, Kristina Lerman, Steven Minton Ion Muslea (2001) A Machine Learning Approach to Accurately and Reliably Extracting Data from the Web IJCAI-2001 Workshop on Text Learning: Beyond Supervision, Seattle
13. Craig A. Knoblock, Steven Minton, Jose Luis Ambite, Naveen Ashish, Pragnesh Jay Modi, Ion Muslea, Andrew G. Philpot, Sheila Tejada (1998) Modeling Web Sources for Information Integration, Proc. Fifteenth National Conference on Artificial Intelligence.
14. Cunningham H., Maynard D., Bontcheva K. and Tablan V.(2002) GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications. In Proc. of the 40th Anniversary Meeting of the Association for Computational Linguistics.
15. Danger R., Berlanga R., (2009) Generating Complex Ontology Instances from Text Documents, Journal of Algorithms, vol.64, Issue 1, Jan., pp 16-30.
16. Davalcu H., Vadrevu S., Nagarajan S. (2003) OntoMiner: Bootstrapping and populating ontologies from domain specific web sites. IEEE Intelligent Systems 18(5) 24-33.
17. De Maneffe M.C.D., Manning C.D (2008) Stanford Typed Dependencies, Manual
18. Dean T., Cordy J., Schneider K., Malten A. (2001) Experience using design recovery techniques to transform legacy systems, Proceeding of 17th International Conference on Software Maintenance, 622-631
19. Dowell L. K., Cafarella M.J. (2006) Ontology –driven Information Extraction with OntoSyphon, International Semantic Web Conference.
20. M.Dzbor, J. Domingue, E. Motta, (2003) Magpie-toward a semantic web browser, Proc. of the International Semantic Web Conference.
21. Ecom J.H.,Zhang B.T., (2004) PubMiner: Machine Learning-based Text Mining for Biomedical Information Analysis. Artificial Intelligence: Methodology, Systems, Applications
22. Han H., Giles C.L., Manavoglu E., Zha H., (2003) Automatic Document Metadata Extraction using Support Vector Machines, Proceedings of the 3rd ACM/IEEE-CS joint conference on Digital libraries, Houston, Texas.
23. Handchuth S., Staab S, Ciravenga F., (2002) S-CREAM – Semi-automatic CREAtion of Metadata, The 13th International Conference on Knowledge Engineering and Management

24. Handchuth S., Staab S, Maedche A., (2001) CREAM – Creating Relational Metadata with a component-based ontology driven framework, Proceedings of K-Cap.
25. Hearst M.,(1992) Automatic acquisition of hyponyms from large text corpora, in proceedings of the 14th international conference on Computational Linguistics
26. Heflin, J. (2001) Towards the Semantic Web: Knowledge Representation in a Dynamic, Distributed Environment. Ph.D. Thesis, University of Maryland, College Park..
27. Heflin, J. and Hendler, J.(2000) Dynamic Ontologies on the Web. In Proceedings of the Seventeenth National Conference on Artificial Intelligence (AAAI-2000). AAAI/MIT Press, Menlo Park, CA, 2000. pp. 443-449.
28. Heflin, J., Hendler, J., and Luke, S. Applying Ontology to the Web: A Case Study. In: J. Mira, J. Sanchez-Andres (Eds.), International Work-Conference on Artificial and Natural Neural Networks, IWANN'99. Proceedings, Volume II. Springer, Berlin, 1999. pp. 715-724.
29. Huck G., Fankhauser P., Aberer K., Neuhold E.J., (1998) JEDI: Extracting and Synthesizing Information from the Web, Conference on cooperative information systems CoopIS, New York
30. Iria José and Ciravegna Fabio (2005) Relation Extraction for Mining the Semantic Web, Proceedings Machine Learning for the Semantic Web Dagstuhl Seminar 05071, Dagstuhl, DE.
31. Iria José (2004) Relation Extraction for Mining the Web, Transfer Report.
32. Iannone L., Palmisano I. and Fanizzi.N.(2007) An algorithm based on counterfactuals for concept learning in the Semantic Web. Applied Intelligence, 26(2), pp. 139--159, Springer Science + Business Media, LLC..
33. Kiyavitskaya N., Zeni N., James R., Mich L., Mylopoulos J., (2005) Semi-Automatic Semantic Annotations for Web Documents, Proceedings of "SWAP 2005"
34. Lavrac N., Dzeroski S., (1994) Inductive Logic Programming: Techniques and Applications, Ellis Horwood, New York.
35. Maynard D., Funk A., Peters W., (2009), SPART: A Tool for Automatic Semantic Pattern-based Ontology Population, Proc. Of International Conference for Digital Libraries and the Semantic Web.

36. Mooney J.R., Nahm U. N., (2002) Text Mining with Information Extraction, Proceedings of AAAI-2002 Spring Symposium on Mining Answers from Texts and Knowledge.
37. Papov B., Kiryakov A., Manov D., Kirilov A., Ognianoff M., Goranav M., Towards Semantic Web Information Extraction Available: <http://gate.ac.uk/conferences/iswc2003/proceedings/popov.pdf>
38. Poon H., Domingos P., (2010) Unsupervised Ontology Induction from Text, ACL' Proc, 48th Annual Meeting of the Association for Computational Linguistics, pp 296-305.
39. Quinlan J.R., Cameron-Jones R.M. (1993) FOIL: A midterm report, In Proceedings of the European Conference on Machine Learning, Vienna, Austria.
40. Sabhashin R., Akilandeswari, J. (2011) A survey on ontology construction methodologies. International Journal of Enterprise Computing and Business System, 1(1).
41. Sahuguet A, Azavant F. (1999) WysiWyg Web Wrapper Factory(W4F), Proceedings of WorldWideWeb Conference, Toronto Ont.
42. Sahuguet A., Azavant F.(1999), Building light-weight wrappers for legacy Web Data-Sources using W4F, International Conference on Very Large Databases, UK. ,
43. Sanchez D. (2008) Domain Ontology Learning from the Web, Ph.D. Thesis, Technical University of Catalonia, Spain.
44. Snoussi, H., Magnin, L. and Nie, J.Y. (2002) Toward an Ontology-based Web Data Extraction, The AI-2002 Workshop on Business Agents and the Semantic Web.
45. Soderland S., Fisher D., Asfltn J., Lehnert W. (1995) CRYSTAL: Inducing a Conceptual Dictionary. Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence, pages 1314 -- 1319
46. Vargas-Vera M., Motta E., Domingue J., Buckingham S., Lanzoni S., Lanzoni M., Knowledge Extraction by using an Ontology-based Annotation Tool.
47. Vargas-Vera M., Mottas E., Dominique M., Stutr F., Ciravenga F., (2002) MnM: Ontology Driven Semi-Automatic and Automatic Support for Semantic Markup, Proceedings of the 13th International Conference on Knowledge Engineering and Management.
48. Wang T., Bontcheva K., Li Y., Cunningham H., (2005) D2.1.2. Ontology Based Information Extraction, SEKT Deliverable D2.1.2. Available: <http://sekt-project.org/rd/deliverables/index.html/>.

49. Wang Y., Hodges J., Tang B. (2003) Classification of Web Documents using a Naïve Bayes Method, Proceedings of the 15th IEEE International Conference on Tools with Artificial Intelligence.
50. Yildiz, B., Miksch S., (2007). Motivating Ontology-Driven Information Extraction, Proceedings of the International Conference on Semantic Web and Digital Libraries (ICS-D-2007)
51. Drumond L., Girardi R, (2010) An experiment using Markov Logic network to extract ontology concepts from the text, in ACM Special Interest Group on Applied Computing, pp. 1354-1358.
52. Richardson M., Domingo P., (2006) Markov Logic Network, Machine Learning, vol. 62, pp 107-136.
53. Mausam, Schmitz M., Bart R., Sonderland S., Etzioni O., (2012) Open Language Learning for Information Extraction, Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, Korea, pp 523-534.
54. Fader A., Soderland S., Etzioni O., (2011) Identifying Relations for Open Information Extraction, Proceedings of EMNLP
55. Etzioni O., Fader A., Christensen J., Sonderland S., Mausam, (2011) Open Information Extraction: the second generation, Proceedings of the International Joint Conference on Artificial Intelligence (IJCAL'11).
56. Wu F., Weld D. S., (2010) Open Information Extraction Using Wikipedia, Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL '10).
57. Carlson A., Betteridge J., Wang R. C., Hruschka Jr. E., R., Mitchell T., (2010) Coupled Semi-supervised Learning for Information Extraction, Proceedings of International Conference on Web Search and Data Mining (WSDM'10).
58. Choi E., Malinowski T., Telemeter L. S., (2015) Scalable Semantic Parsing with Partial Ontologies, Proceedings of Association of Computational Linguistics.
59. Mintz M., Bills S., Snow R., Jurafsky D., (2009). Distant supervision for relation extraction with-out labeled data. Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Vol. 2, pp 1003–1011, Morristown, NJ, USA. Association for Computational Linguistics.

60. Yao L., Riedel S., McCallum A., (2010) Collective Cross-Document Relation Extraction without Labelled Data, Proceedings of the Conference on Empirical Methods in Natural Language Processing, USA, pp: 1013-1023, October.
61. Yao L., Haghighi A., Riedel S., McCallum A., (2011) Structured Relation Discovery Using Generative Models, Proceedings of the Conference on Empirical Methods in Natural Language Processing, Scotland, pp: 1456-1466, July
62. Pawar S., Bhattacharya P., Palshikar G. K., (2016) End-to-End Relation Extraction using Markov Logic Network, Proceedings of the 17th International Conference on Intelligent Text Processing and Computational Linguistics.
63. Riedel S., Yao L., McCallum A., Marlin B. M., (2013) Relation Extraction with Matrix Factorization and Universal Schemas, Proceedings of NAACL-HLT, pp:74-84.
64. McCallum A., Nigam K., (1998) A Comparison of Event Models for Naive Bayes Text Classification, AAAI-98 Workshop Learning for Text..
65. Rennie J., Shih L., Teevan J., Karger D., (2003) Tackling Poor Assumptions of Naive Bayes Text Classifiers, ICLM.
66. Tang B., He H., Bagenstoss P.M., Kay S., (2016) A Bayesian Classification Approach using Class-specific Features for Text Categorization, IEEE Transactions on Knowledge and Data Engineering, 28, No 6, pp. 1602-1606
67. Joachims H., (1998) Text Categorization with Support Machines: Learning with many Relevant Features, ECML-98..
68. Furey T.S., Christianini N, Duffy N, Bednarski D.W., Shummer M., Haussier D., (2000) Support Vector Machine Classification and Validation of Cancer Tissue Samples Using Microarray Expression Data, Bioinformatics, vol. 16, pp. 906-914.
69. Drucker H., Wu D., Vapnik V.N., (1999) Support Vector Machines for Spam Categorization, IEEE Transactions on Neural Networks, vol. 10, No. 5, pp 1048-1054.
70. Joachims T., (1996) Probabilistic Analysis of the Rocchio Algorithm with TFIDF for Text Categorization, Proceedings of Int. Conference on Machine Learning, pp. 143-151.
71. Lewis D. D., Schapore R. E., Call J. P., Papka R., (1996) Training Algorithms for Linear Text Classifiers, Proceedings of 19th Int. ACM SIGIRI Conference Research and Development in Information Retrieval, pp. 298-306..

72. Cunningham P., Delany S.J., K-Nearest Neighbour Classifiers, (2007) Dublin Technical Report, UCD-CSI-2007-4..
73. Tan S., Cheng X., (2007) An Effective Approach to Enhance Centroid Classifier for Text Categorization, Proceedings of 11th European Conference on Principles and Practice of Knowledge Discovery in Databases, pp. 581-588.
74. Pang G., Jin H., Jiang S., (2014) CenKNN: Scalable and Effective Text Classifier, Data Mining and Knowledge Discovery, January..
75. Lam W., Han Y., (2003) Automatic Textual Document Categorization Based on Generalized Instance Sets and a Metamodel, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 25, no. 5, May.
76. Apte C., Damerau F., Weiss S.M., (1994) Automated Learning of Decision Rules for Text Categorization, ACM Transactions Information Systems 12, pp. 233-252, July.
77. Weiss S.M., Indurkha N., (1993) Optimized Rule Induction, IEEE Exp. 8(6), pp. 61-69.
78. Zaiane O., Antonie M., (2002) Classifying Text Documents by Associating Terms with Text Categories, Proceedings of 13th Australasian Database Conference(ADC'02), Melbourne, Australia, pp. 215-222.
79. Haralambous Y., Lenca P., (2014) Text Classification Using Associative Rules, Dependency Prunning and Hyperonymization, Proceedings of DMNLP, Nancy, France, CEUR Workshop Proceedings, vol. 1202, pp. 65-80.
80. Riloff E., Lehnert W., (1994) Information Extraction as a Basis for High-precision Text Classification, ACM Transactions of Information Systems, vol. 12, issue 3, pp 296-333.
81. Agrawal C. C. , Zhai C. X., (2012) A Survey of Text Classification Algorithms in Mining Text Data, Chap. 6, Springer, pp: 163-222.
82. Apte C. , Damerau F., Weiss S.M., (1994) Automated Learning of Decision Rules for Text Categorization, ACM Transactions on Information Systems.
83. Yih W., Chang M., He X., Gao J., (2015) Semantic Parsing via Staged Query Graph Generation: Question Answering with Knowledge Base, Proceedings of Association for Computational Linguistics(ACL)..
84. Chiticariu L., Li Y., Reiss F. R., (2013) Rule-based Information Extraction is Dead! Long Live Rule-based Information Extraction Systems., Proceedings of the Conference on Empirical Methods in Natural Language Processing, Washington, pp : 827-832 .

85. Singla P., Domingos P., (2005) Discriminative training for Markov Logic Network, AAAI-2005, pp 868-873.
86. Lowd D. , Domingos P., (2007) Efficient Weight Learning for Markov Logic Networks, PKDD-7, pp 200-211.
87. Shewchuck J., An Introduction to the Conjugate Gradient Method without the Agonizing Pain, Technical Report, CMU-CS-94-125, School of Computer Science, Carnegie Mellon University, 1994.
88. <http://www.icaen.uiowa.edu/~comp/Public/Apriori.pdf>
89. http://en.wikipedia.org/wiki/Ontology_%28information_science%29#Ontology_components
90. <http://www.geneontology.org/GO.doc.shtml>
91. <http://protege.cim3.net/file/pub/ontologies/camera/camera.owl>
92. <http://www.co-ode.org/ontologies/amino-acid/2005/10/11/amino-acid.owl>
93. <http://www.w3.org/TR/owl-guide/>
94. <http://www.daml.org/2001/03/daml+oil-index.html>
95. <http://www.w3.org/TR/rdf-schema/>
96. <http://svmlight.joachims.org/>
97. <https://midas.psi.ch/elog/>
98. http://en.wikipedia.org/wiki/Support_vector_machine
99. <http://jedlik.phy.bme.hu/~gerjanos/HMM/node2.html>
100. OntoWeb Deliverable 1.3: A Survey on Ontology Tools (2002)
<http://icc.mpei.ru/documents/00000826.pdf>
101. http://www.itl.nist.gov/iaui/894.02/related_projects/muc/proceedings/ne_task.html
102. http://en.wikipedia.org/wiki/Supervised_learning
103. http://en.wikipedia.org/wiki/Unsupervised_learning
104. <http://sisla06.samsi.info/jpal/mult1031.pdf>

- 105. http://en.wikipedia.org/wiki/Precision_and_recall
- 106. <http://nlp.stanford.edu/software/lex-parser.shtml>
- 107. <http://www.itl.nist.gov/iad/mig/tests/ace/2004/>

Appendix A

Fundamentals of Ontology

An ontology is theoretically defined as a formal explicit specification of a shared conceptualisation.[52] It can be described as a formal representation of knowledge of entities within a domain and relationships between these entities, necessary to understand the underlying information.

Ontology may contain the following components

Entities/Classes – domain specific concepts

Instances- individual elements of a class

Relations – ways in which entities can be related to one another.

Properties – aspects, features, characteristics and parameters that an entity can have.

Restrictions – formally stated statement that is used to test the validity of an assertion in accepting as an entity.

Inference Rules - statements in the form of if-then sentences which are used to draw logical expression from the ontological information.

Events- changing of entities and relations.

Irrespective of the way and the language used for formal representation of knowledge, almost all ontologies describe entities and relations. Properties of an entity can also be extracted in the form of a relation. But ontologies vary greatly in size, scope and semantics. Ontologies can range from generic upper ontologies to domain specific ontologies. As the name implies domain ontology represents a specific domain by describing the terms within the domain. Medical ontologies, Camera ontology are examples for domain ontology.

An upper ontology represents common objects such as time, event, person etc. that are generally applicable across a wide range of domain ontologies. Concepts introduced in an upper ontology can be specialized in building domain ontologies. GFO, CYC, SUMO WordNet are considered as examples for upper ontology.

Ontology Engineering refers to the activities that concerns construction of ontologies, methods and methodologies for ontology construction, ontology life cycle, tools and languages that support them. “It aims to make explicit the knowledge contained within software applications and within enterprises and business procedures for a particular domain. Ontology Engineering offers a direction towards solving the interoperability problems brought about by semantic obstacles, such as obstacles related to the definitions of business terms and software classes”. Ontologies can be created by knowledge representation experts or novice web users differing widely in style and semantics. There are large ontologies which cover numerous concepts and relations and small ontologies with handful of concepts. Ontology Engineering plays a significant role in such a diverse and heterogeneous information space.

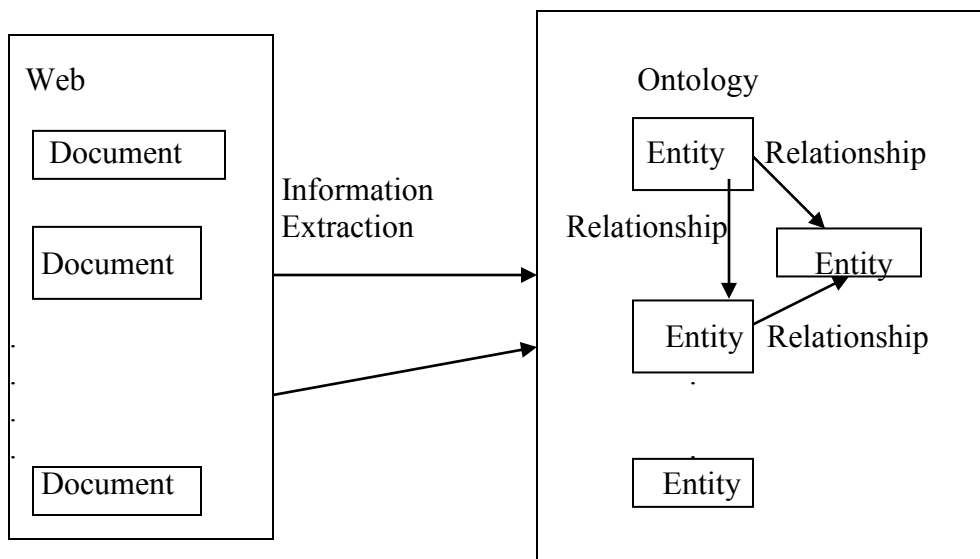


Fig. 1 Abstract view of the ontology development

Appendix B

Table B.1 - Summary of the related work on Information Extraction

System Ontology based	Techniques Used	Outcome	Advantages	Limitations & Disadvantages
WebKb [11]	Machine learning algorithm Foil and statistical method Naïve base for extracting class instances. Sequence Rule Validation for extraction of Text fields.	Extraction rules for class instances and relations. Extracted information in the knowledge base.	Yields satisfactory results within the limitations.	Class instances are restricted to whole web page, Known relations are only identified. Negative instances are extracted.
WeDax [44]	An agent that uses a manually constructed definition using schema for object oriented XML.	Extraction of data for population of ontology.	Capability of the agent in converting HTML documents into XML formats or extracting data from various sources.	Assumption of the rigid web page structures. Involvement of the developer in the manual creation of descriptions in data extraction when adapting the system to different domains.
OntoSho pie [5]	Natural Language Processing system Marmot for processing the document. The dictionary	Generation of concept definitions to populate an ontology	Selects best rules by assigning a confidence value. Work well in specific domains.	Is not customized. Heavy developer/user involvement in adapting to different domains

	induction system Crystal for generation of extraction rules			
OntoSyn hon [19]	Five Hearst phase template	Extraction of sub class instances	Associated learning figure to improve the accuracy of extracted instances	Limited usage. Only taxonomical relations are dealt
Amilcare [8]	LazyNLP and (LP) ² algorithm	Extraction rules	Easily integrated to any system	Relation extraction is not performed
Armadillo [10]	Rule generation for wrappers.	Extracted information in the form of Subect- Verb_Object for document annotation	Easily switched to various domains. Unannotated portion is further processed. Can address issues of some syntactic and semantic ambiguities.	Basic terms are extracted. Extracting information from complex sentences is not demonstrated.
T-Rex [30,[31]	Continuation of Amilcare	Framework for developing algorithms for relation Extraction.		
Burcu Yildiz & Silvia Miksch's work [50]	Pattern learning using bag of words and neighbourhood words	Extraction Rules	Ontology Management module to adapt the system to different domains.	Not capable of identifying non- taxonomical relations.

Borisfav Popav and the team's work [37]	GATE for developing rules. Semantic Gazetteer	Extraction of Name entities	Use of pre populated knowledge base to verify the extracted name. Identification of the variation of the same term.	Relation extraction is not performed. Name entity reference ambiguities still exists.
Text2Onto [7]	GATE for developing rules	Instances and relations for an ontology initiation model.	Translation of extracted information in Probabilistic Ontology Model (POM) to any ontology language. Filtration of irrelevant information	Identification of is-a ,part-of and some general relations only.
Roxana Danger and Rafeal Berlanga [15]	Similarity function and lexical description from ontology for entity instances Segment scope definition and inference rule in ontology for relation extraction	Entity instances and limited relations	Use of entity recognizers and disambiguators to find the initial set of instances.	Association of scope definition to text segments according to a hierarchical document structure.
OntoUSP [38]	Markov logic to extract relations	Hierarchical relations	Use of unsupervised semantic parsing	Concentrates only hierarchical relations.
SPART [35]	Plug-in to GATE, Linguistic patterns	Entities and Relations	Research platform which enables	

	including Hearst.		addition of any kind of knowledge.	
Atiken[1]	Foil Algorithm	Entities and relations	Any type of relations are considered.	Success of the system depends on selected training data
R. J. Mooney and Nahm U. N. [36]	IPL Techniques	Relations	Targeted at complicated relations	Domain specific approach
David Sanchez's research [43]	Linguistic patterns such as Hearst patterns, noun phrased based patterns. Unsupervised learning	Entities and relations for domain ontologies	Verification of extracted information., extraction of non taxonomical relations	Restricted to only limited sentence structures for non taxonomical relation extraction.
Robert Bauring and the team's work [2]	The tool Lixto for wrapper generation	Ontology population with extracted information	Focused in adapting their system to various application domains.	Extraction of basic terms only
SHOE [26]	Developed tools in JAVA.	Annotation of HTML pages with SHOE.	A complete system with wide applicability	The user should be willing to do the additional work with SHOE annotation.
OntoLT [3]	A language is provided for the user to define mapping rules predefined mapping rules are included in	Extraction or extension of Protégé ontology with extracted	Provides an environment for the user to experiment with.	Concentrated more on noun phrases.

	OntoLT. Chi-square function is used to determine domain specific terms.	entities.		
OntoMiner [16]	Semantic Partitioning	Ontology	Ontology is developed from the scratch	Semantically related terms are not identified.
Choi E. et al [58]	Semantic parsing model	Entities	Semi supervised approach. Strong performance on entity attribute extraction	Effective on highly subjective entities.
Systems not based on an ontology				
Ariadne [13]	Machine learning to induce rules for wrappers	Produce answers for user queries	Integration of web sites to gather information for user queries.	Many training examples and heavy user involvement.
STALKE R [12]	Machine learning to induce rules for wrappers	Produce answers for user queries	Requires a only small numbers of examples	Can make use same structured web pages.
DIScoTeX [36]	RAPIER and BWI algorithms for information extraction. KDD techniques RIPPER and APRIORI for induction rules to find additional facts.	Extract instances to fill values for the slots of a particular entry.	Effective performance on short web pages of similar structure.	Can't distinguish among different occurrence of the same term. Needs higher number of training examples.

PubMiner [21]	Part of Speech(POS) tagger based on Hidden Markov Model for tagging words. Named Entity Tagger based on support vector machines to recognize a region of an entity. Association Rule Discovery using Apriori algorithm for generation of rules.	Extraction of entities and relations from biological literature	Both entities and relations are extracted from a massive literature	Extracts false positives in the domain. Restricted to Biological domain.
Kiyavitskaya N. et al [33]	Methodology from LS/2000 software analysis to mark up the document	Documents marked up with XML grammar		Only basic entities are identified.
Han H. et al [22]	Support Vector Machines and implemented using SVMlight.	Classify header words in research papers into 15 classes. Identify chunk boundaries in multi class lines.		Assume that each line has only one chunk boundary.
Drummond et al [51] (Rule based) Supervised learning	tf-idf measure to filter and extract the noun and noun phrases to use in rules Markov logic network to find the probability of extracted concept	Concepts	Incorporating different techniques in a multi-rule system	No reliable implications for generalization

OLLIE (Training set is found by bootstraping) [55]	Open Information extraction, Learn patterns from sentences which consists of the entity instances and the relation verb of the high precision tuples obtained from ReVerb system	Relation instances	Is able to consider wide range sentence structures including some complicated sentence structures for accurate relation extraction	Constraint impose to correct the assumption made that any sentence which contains entities and the relation in the seed tuple is a candidate for the relation, may not work in any type of sentence.
Carlson et al (semi supervised) [57]	Use seed instances and patterns with some constraints.	Instances for concepts and relations	Semi supervised nature and not restricted to pre defined relations	Constraints used may not be sufficient for all kind of relations. Low precision for some relations.
Mints et al [59] (distance supervision)	Use relations in freebase to extract lexical and syntactic features of feature vector	Relations	Not requiring a large set of labeled data	Low precision on some relations. Getting noisy features added to the feature vector
Yao et al [60] (distance/weak supervision)	Use set features from text corpus for freebase Relations and rank the extracted relation instances using the MAP state in the Markov Network environment. Follows few other works.	Relations	Claims increased precision in one of the experiment text domain over isolated baseline approach and pipeline approach	Precision depends on the availability of the relations in the freebase. Unknown relations are not identified.

Yao and Haghighi at el [61] (Unsuper vised approach)	based on Latent Dirichlet Allocation(LDA) on topic model	Relations		
Pawar et al [62]	Use rule based three maximum entropy classifiers. Rules are modeled in Markov Logic environment.	Entities and relations		Entity types and relation types are restricted to ACE 2004 entity and relation types. Proper weight learning methods is not used.
Riedel et el [63]	based on extensions of probabilistic models of matrix factorization and collaborative filtering	Relations	claim that their model can predict relations which are not existing in the Freebase.	Computational Complexity

Appendix C

ANNIE- GATE's Information Extraction System

ANNIE is armed with the following resources which are automatically loaded with it. All these resources should be used together in the process of annotating a document with relevant entities.

Document Reset

The document reset resource enables the document to be reset to its original state by removing all the annotated states and their contents, apart from the one containing document format analysis (original markups).

ANNIE Tokeniser

Tokeniser is the component bundled in ANNIE to split the text into very simple tokens such as words, numbers, punctuation etc. Tokeniser uses tokeniser rules which distinguishes between uppercase and lowercase letters and identifies the type of the token. Tokeniser rules can identify token types; word, number, symbol, punctuation and space. English Tokeniser comprises a normal tokeniser and a jape transducer which has the role of adapting the generic output of the tokeniser to the requirement of English part of speech tagger.

Gazetteer

The Gazetteer contains lists of names such as surnames, names of countries, names of organizations etc as a plain text file and an index file to access these lists. The Gazetteer lists are created with one entry per line. Index file specifies a major type and optionally a minor type. Gazetteer lists are compiled into finite state machines. Any text token matched by gazetteer list according to the grammar rules, will be annotated with features specifying the major and minor types. For an example in the following entry of the index file, the first column refers to the list name “government”, the second column to the major type “organization” and the third to the minor type “government”.

government.list : organization : government

A part of the corresponding gazetteer list is shown below.

Advanced Research Projects Agency

Aeronautica Civil

Air Force

AIR FORCE

Army

ARMY

Army Corps

Army Corps of Engineers

A.S.I.

ASI

Cabinet

Canada Post

Canadian Space Agency

Grammar rules specify the types, major or minor to be identified for a token matched in particular circumstances. The token will then be annotated with features specifying the major and minor types.

Sentence Splitter

The Sentence Splitter segments the text into sentences using a gazetteer list of abbreviations which helps to distinguish full stops from other characters. Each sentence is annotated with the type Sentence. The sentence splitter is domain and application independent.

RegEx Sentence Splitter

The aim of the RegEx Splitter is to address some performance issues in the jape based splitter mainly to do with improving the execution time and robustness specially with irregular inputs. The RegEx Splitter provide facilities for

internal splits – sentence splits which are part of a sentence such as sentence ending punctuation.

external splits – sentence splits which are not part of a sentence such as two consecutive lines

non splits – text fragments which are not sentence splits such as full stops occurring inside abbreviation.

Part of Speech Tagger

Part of Speech is a linguistic category of words which is generally defined by the syntactic or morphological behaviour of the lexical item. Common linguistic categories include noun, verb, determiner etc. Pos tagging is the process of marking up the words in a text as corresponding to a particular part of speech based on both its definition as well as its context. There are well established part of speech taggers available to identify above mentioned lexical items. ANNIE uses the Hepple tagger(ref) which produces a part-of –speech tag as an annotation on each word or symbol. The tagger uses default lexicon and a set of rules. When using, the default lexicon is replaced by the appropriate lexicon at the time the tagger is loaded.

Semantic Tagger (ANNIE NE Transducer)

Semantic Tagger uses rules built in JAPE in order to identify semantic categories which are considered as entities in a domain. Semantic tagger contains rules to identify general entities such as Person, Location, Job Title etc. in order to produce outputs of annotated entities. ANNIE provides facilities to build additional rules for the purpose of identifying domain specific entities and incorporate new rules to the Semantic Tagger. ANNIE' Semantic Tagger provides a set of annotations which represent the most general entities in a text and annotations appear at the GATE user interface according to their occurrence in the text. The annotation set given by ANNIE includes the following set of entities.

{Address, Date, FirstPerson, JobTitle, Location, Lookup, Organization, Person, Sentence, SpaceToken, Split, Title, Token, UnKnown, UrlPre, Percent, Temp, Identifier, Money}.

Orthographic Coreference (OrthoMatcher)

OrthoMatcher module adds identity relations between named entities found by the semantic tagger in order to perform coreference. It may assign a type to an unclassified proper name or pronoun, using the type of a matching name.

When ANNIE' is used for entity extraction Tokenizer should be run on the text .before NE Transducer is used. New entity extraction rules can be accommodated in GATE by modifying ANNIE or adding new processing resources. GATE's extendibility permits domain specific entities to be added to the annotation set through new extraction rules and extraction rules can be compiled into a new processing resource

Appendix D

JAPE rules and creole for the entity extraction in domain Bird

Phase: Colour
Input: Lookup Token
Options: control = appelt

```
Rule: ColourId
(
  {Lookup.majorType == colours}
)
:color -->
  : color.Colour = {rule = "ColourId"}
```

```
/*
 * main.jape
 *
 * Copyright (c) 1998-2004, The University of Sheffield.
 *
 * This file is part of GATE (see http://gate.ac.uk/), and is free
 * software, licenced under the GNU Library General Public License,
 * Version 2, June 1991 (in the distribution as file licence.html,
 * and also available at http://gate.ac.uk/gate/licence.html).
 *
 * Diana Maynard, 02 Aug 2001
 *
 * $Id: main.jape 5921 2004-07-21 17:00:37Z akshay $
 */
```

MultiPhase: TestTheGrammars
Phases:
color
Measurement

Phase: Measurement
Input: Lookup Token SpaceToken
Options: control = appelt

```
Macro: Measure1
(({Token.kind == number}
  ({Token.string == "."}
    {Token.string == number}))?)
({SpaceToken}
  ({Token.string == "cm"}|
  {Token.string == "m"}|
  {Token.string == "'"}))
)
)
```

```
Macro: Weigh
(({Token.kind == number}
  ({SpaceToken}
    {Token.string == "kg"}|
```

```

    {Token.string == "Kg"}|
    {Token.string == "g"}|
    {Token.string == "oz"})
)

Rule: Length1
(
    (Measure1)
    :length
    //({SpaceToken}
    //{Token.string == "in"}
    // {Token.string == "length"}
    //)
)
-->
    :length.Length = {kind = "length", rule = "Length1"}

Rule: WeightX
(
    (Weigh)
    :weight
    //({Token.kind == space}
    // {Token.string == "in"}
    // {Token.string == "weight"}
    // )
)
-->
    :weight.Weight = {kind = "weight", rule = "WeightX"}

Rule: BirdName
(
    {Lookup.majorType == birds}
)
:bird -->
    : bird.Bird = {kind = "bird", rule = "BirdName"}

Rule: FamilyName
(
    {Token.string == "family"}
    {SpaceToken}*
)
(
    {Token.category == NN}|{Token.category == NNS}
)
:family -->
    : family.Family = {kind = "family", rule = FamilyName}

Macro: DietOf

```

```
(
  {Token.string == "diet"}
  {SpaceToken}
  ({Token.string == "of"}
   {SpaceToken})?
  ({Bird}
   {SpaceToken})?
)
```

Macro: DietSources

```
(
  {Token.string == "food"}
  {SpaceToken}
  {Token.string == "sources"}
  {SpaceToken}
)
```

Macro: MoreDiet

```
(
  {Token.string == ","}
  {SpaceToken}
)
```

Macro: AndDiet

```
(
  {Token.string == "and"}
  {SpaceToken}
)
```

Rule: Diet1

```
(
  {DietOf}
  ({Token.category == RB}
   {SpaceToken})?
  {Token.string == "consist"}
  {SpaceToken}
  {Token.string == "of"}
  {SpaceToken}
)
(
  {Token.category == NN}
)
:diet1 -->
:diet1.Diet = {kind = "diet1", rule = "Diet1"}
```

Rule: Diet2

```
(
  {DietOf}
```

```

    {Token.category == VB}
    {SpaceToken}
  )
  (
    {Token.category == NN}
  )
:diet2 -->
  :diet2.Diet = {kind = "diet2", rule = Diet2}

```

Rule: Diet3

```

  (
    {Token.string == "feed"}
    {SpaceToken}
    ({Token.category == RB}
     {SpaceToken})?
    {Token.string == "on"}
    {SpaceToken}
  )
  (
    {Token.category == NN}
  )
:diet3 -->
  :diet3.Diet = {kind = "diet3", rule = Diet3}

```

Rule: Diet4

```

  (
    {Token.string == "eat"}
    {SpaceToken}
  )
  (
    ({Token.category == JJ}
     {SpaceToken})?
    {Token.category == NN}
  )
:diet4 -->
  :diet4.Diet = {kind = "diet4", rule = Diet4}

```

Rule: Diet5

```

  (
    {DietSources}
    {Token.string == "such"}
    {SpaceToken}
    {Token.string == "as"}
    {SpaceToken}
  )
  (
    {Token.category == NN}
  )
:diet5 -->
  :diet5.Diet = {kind = "diet5", rule = Diet5}

```

Rule: Diet6

```
(
  {DietSources}
  {Token.string == "including"}
  {SpaceToken}
)
(
  {Token.category == NN}
)
:diet6
(
  {Token.string == ","}
  {SpaceToken}
)*
:diet6 -->
  :diet6.Diet = {kind = "diet6", rule = Diet6}
```

Macro: DetailHabitat

```
(
  ({Token.category == JJ}
   {SpaceToken})?
  ({Token.category == VB}
   {SpaceToken})?
  {Token.category == NN}
)
```

Rule: Habitat1

```
(
  {Token.string == "inhabit"}
  {SpaceToken}
)
(
  {DetailHabitat}
)
:habitat1 -->
  :habitat1.Habitat = {kind = "habitat1", rule = Habitat1}
```

Rule: Habitat2

```
(
  {Token.string == "habitat"}
  {SpaceToken}
  {Token.string == "is"}
  {SpaceToken}
)
(
  {Token.category == VB}?

```

```

    {DetailHabitat}
  )
: habitat2 -->
  :habitat2.Habitat = {kind = "habitat2", rule = Habitat2}

```

Rule: Eggs1

```

(
  {Token.string == "lay"}
  {SpaceToken}
  ({Token.string == "up"}
   {SpaceToken}
   {Token.string == "to"}
   {SpaceToken}
  )?
  (
    {Token.kind == number}
  )
: egg1
(
  ({SpaceToken}
   {Colour}))?
  {SpaceToken}
  {Token.string == "eggs"}
)
-->
:egg1.EggNo = {kind = "egg1", rule = Eggs1}

```

Rule: Eggs2

```

(
  {Token.string == "eggs"}
  {SpaceToken}
  ({Token.category == RB}
   {SpaceToken}))?
)
(
  {Token.kind == number}
)
:egg2 -->
:egg2.EggNo = {kind = "egg2", rule = Eggs2}

```

Rule: Eggs3

```

(
  {Token.kind == number}
)?
: egg3
(
  {SpaceToken}
  {Token.string == "to"}
  {SpaceToken}
)?
(
  {Token.kind == number}

```

```
)  
: egg3 -->  
(  
  {SpaceToken}  
  {Token.string == "eggs"}  
)  
:egg3.EggNo = {kind = "egg3", rule = Eggs3}
```


Appendix E

Part-of-Speech Tags used in the Hepple Tagger

CC - coordinating conjunction: "and", "but", "nor", "or", "yet", plus, minus, less, times (multiplication), over (division). Also "for" (because) and "so" (i.e., "so that").

CD - cardinal number

DT - determiner: Articles including "a", "an", "every", "no", "the", "another", "any", "some", "those".

EX - existential there: Unstressed "there" that triggers inversion of the inflected verb and the logical subject; "There was a party in progress".

FW - foreign word

IN - preposition or subordinating conjunction

JJ - adjective: Hyphenated compounds that are used as modifiers; happy-go-lucky.

JJR - adjective - comparative: Adjectives with the comparative ending "-er" and a comparative meaning. Sometimes "more" and "less".

JJS - adjective - superlative: Adjectives with the superlative ending "-est" (and "worst"). Sometimes "most" and "least".

JJSS - -unknown-, but probably a variant of JJS

-LRB- - -unknown-

LS - list item marker: Numbers and letters used as identifiers of items in a list.

MD - modal: All verbs that don't take an "-s" ending in the third person singular present: "can", "could", "dare", "may", "might", "must", "ought", "shall", "should", "will", "would".

NN - noun - singular or mass

NNP - proper noun - singular: All words in names usually are capitalized but titles might not be.

NNPS - proper noun - plural: All words in names usually are capitalized but titles might not be.

NNS - noun - plural

NP - proper noun - singular

NPS - proper noun - plural

PDT - predeterminer: Determinerlike elements preceding an article or possessive pronoun; "all/PDT his marbles", "quite/PDT a mess".

POS - possessive ending: Nouns ending in "'s" or ""'".

PP - personal pronoun

PRPR\$ - unknown-, but probably possessive pronoun

PRP - unknown-, but probably possessive pronoun

PRP\$ - unknown, but probably possessive pronoun, such as "my", "your", "his", "his", "its", "one's", "our", and "their".

RB - adverb: most words ending in "-ly". Also "quite", "too", "very", "enough", "indeed", "not", "-n't", and "never".

RBR - adverb - comparative: adverbs ending with "-er" with a comparative meaning.

RBS - adverb - superlative

RP - particle: Mostly monosyllabic words that also double as directional adverbs.

STAART - start state marker (used internally)

SYM - symbol: technical symbols or expressions that aren't English words.
 TO - literal to
 UH - interjection: Such as "my", "oh", "please", "uh", "well", "yes".
 VBD - verb - past tense: includes conditional form of the verb "to be"; "If I were/VBD rich...".
 VBG - verb - gerund or present participle
 VBN - verb - past participle
 VBP - verb - non-3rd person singular present
 VB - verb - base form: subsumes imperatives, infinitives and subjunctives.
 VBZ - verb - 3rd person singular present
 WDT - wh-determiner
 WP\$ - possessive wh-pronoun: includes "whose"
 WP - wh-pronoun: includes "what", "who", and "whom".
 WRB - wh-adverb: includes "how", "where", "why". Includes "when" when used in a temporal sense.

:: - literal colon
 , - literal comma
 \$ - literal dollar sign
 - - - literal double-dash
 " " - literal double quotes
 ' ' - literal grave
 (- literal left parenthesis
 . - literal period
 # - literal pound sign
) - literal right parenthesis
 ' - literal single quote or apostrophe

Appendix F

Definitions of Stanford Dependencies

abbrev: abbreviation modifier

- define an abbreviation abbrev(NP, abbreviation)

acom: adjectival complement

- an adjectival phrase that functions as the complement. acomp(verb, adjective)

advcl: adverbial clause modifier

- clause modifying the verb. advcl(verb, modifier)

advmod: adverbial modifier

- adverb or adverbial phrase that serves to modify the meaning of the word.

advmod(word, adverb)

agent: agent

- the complement of a passive verb which is introduced by the preposition “by” and does the action. agent(passive verb, agent)

amod: adjectival modifier

- adjectival phrase that serves to modify the meaning of the NP. amod(noun, adjective)

appos: appositional modifier

- NP immediately to the right of the first NP that serves to define or modify that NP.

appos(noun, noun)

attr: attribute

- WHNP(NP beginning with a wh word such as what, which etc.) complement of a copular verb. Attr(copular verb, WHNP complement)

aux: auxiliary

- non-main verb of a clause. aux(main verb, non-main verb)

auxpass: passive auxiliary

- non-main verb of a clause which contains passive information.

auxpass(passive verb, non-main verb)

cc: coordination

- relation between an element of a conjunct and the coordinating conjunction word of the conjunct. cc(element of conjunct, conjunction word).

ccomp: clausal complement

- dependence clause with an internal subject which functions like an object of a verb or and adjective. ccomp(verb/noun, verb/adjective).

complm: complementizer

- the word introducing a clausal complement.

complm(verb/noun, complementizer)

conj: conjunct

- relation between two elements connected by a coordinating conjunction.

conj(word/number, word/number)

cop: copular

- relation between a complement of a copular verb and the copula verb.

cop(complement, copular verb)

csubj: clausal subject

- clausal syntactic subject of a clause, i.e. the subject of a sentence is itself a clause.

csubj(verb/complement of a verb, verb in the subject clause)

csubjpass: clausal passive subject

- clausal syntactic subject of a passive clause.

dep: dependent

- used when the system is unable to determine more precise dependency relation between two words.

det: determiner

- relation between the head of a NP and its determiner. det(head of NP, determiner).

dobj: direct object

- the noun phrase which is the (accusative) object of the verb in a VP.

dobj(verb, object of the verb)

expl: expletive

- captures an existential “there”. expl(copular verb, there)

infmod: infinitival modifier

- infinitive that serves to modify the meaning of the NP. infmod(noun, modifier)

iobj: indirect object

- the noun phrase which is the (dative) object of the verb in the VP.

iobj(verb, object of the verb)

mark: marker

- the word introducing an adverbial clausal complement. mark(verb, marker)

mwe: multi-word expression

- used for certain multi-word idioms that behaves like a single function word i.e. inside the expressions: rather than, as well as, instead of, such as, because of, in addition to, all but, due to.

neg: negation modifier

- relation between a negation word and the word it modifies.

neg(noun/verb, negation word)

nn: noun compound modifier

- any noun that serves to modify the head noun in NP. nn(noun, noun)

npadvmod: noun phrase as an adverbial modifier

- captures various places where something syntactically a noun phrase and is used as an adverbial modifier in a sentence.

nsubj: nominal subject

- a noun phrase which is the syntactic subject of a clause.

nsubj(verb(not a copular verb), noun)

nsubjpass: passive nominal subject

- a noun phrase which is the syntactic subject of a passive clause.

num: numeric modifier

- any number phrase that serves to modify the meaning of the noun.

num(noun, number)

number: element of compound number

- part of a number phrase or currency amount

parataxis: parataxis

- a relation between the main verb of a clause and other sentential elements, such as sentential parenthetical or a clause after a “;” or a “:”.

partmod: participial modifier

- participial verb form that serves that serves to modify the meaning of a noun phrase or verb phrase. partmod(noun/verb, participial verb)

pcomp: prepositional complement

- head of a clause following the preposition or the preposition head of the following prepositional phrase. pcom(preposition, verb)

pobj: object of a preposition

- head of a noun phrase following the preposition or the adverbs “here” and “there”.

pobj(preposition, noun)

poss: possession modifier

- holds between the head of a NP and its possessive determiner
poss(noun, possessive determiner)

possessive: possessive modifier

- appears between head of a NP and genitive 's. possessive(noun, 's)

preconj: preconjunct

- relation between head of the head of a NP and a word that appears at the beginning a conjunction such as “either”, “both”, “neither” preconj(noun, conjunction)

predet: predeterminer

- the relation between the head of a NP and a word that precedes and modifies the meaning of the NP determiner. predet(noun, determiner)

prep: prepositional modifier

- any prepositional phrase that serves to modify the meaning of the verb, adjective, noun or even another preposition. Prep(verb/adjective/noun, preposition)

prepc: prepositional clausal modifier.

- a clause introduced by a preposition which serves to modify the meaning of the verb, adjective or noun.

pvt: phrasal verb particle

- identifies a phrasal verb and holds between the verb and its particle.

Pvt(verb, particle)

punct: punctuation

- any piece of punctuation in a clause.

purpcl: purpose clause modifier

- is a clause headed by “in order to” specifying a purpose.

quantmod: quantifier phrase modifier

- element modifying the head of a QP(Quantifier Phrase i.e. complex measure/amount used within NP) constituent.

rcmod: relative clause modifier

- relative clause modifying the noun phrase

ref: referent

- relative word modifying the introducing the relative clause modifying the noun phrase. ref(noun, relative word)

rel: relative

- head of the WH-phrase introducing a relative clause.

root: root

- root of the sentence.

tmod: temporal modifier

- noun phrase constituent that serves to modify the meaning of the constituent by specifying a time (i.e. last night, yesterday, tomorrow etc) tmod(verb, time)

xcom: open clausal complement

- clausal complement without its own subject whose reference is determined by an external subject.

xsubj: controlling subject

- relation between the head of a open clausal complement and the external subject of that clause.

Appendix G

A Sample of Reduced Dependencies for the Relations in the Domains Bird and Sport

Domain - Bird

Relation `located_in`(Bird,Location)

```
nsubj(are_native-3, Ostriches-1)
prep_to(are_native-3, savannas-5)
conj_and(are_native-3, Sahel_of_Africa-8)
```

```
nsubj(found-5, Humming_Birds-2)
prep_in(found-5, Cuba-7)
prep_in(Cuba-7, Isle_of_Youth-9)
```

```
nsubj(found-7, diversity_of_Parrots-3)
prep_in(found-7, America-10)
prep_in(found-7, Australasia-12)
conj_and(America-10, Australasia-12)
```

```
nsubj(live-2, Cranes-1)
prep_on(live-2, continents-5)
```

```
nsubj(occurs-5, family_Doves-2)
conj_and(Indomalaya-15, Australasia-17)
```

```
nsubj(have-2, They-1)
prep_in(found-11, Old_World-15)
prep_in(found-11, Australia-17)
conj_and(Old_World-15, Australia-17)
```

```
nsubj(are-2, Penguins-1)
partmod(birds-9, living-10)
prep_in(living-10, Sourthen_Hemisphere-16)
```

```
nsubj(found-3, Parrots-1)
prep_on(found-3, continents-9)
prep_in(continents-9, Australia-11)
conj_and(Australia-11, South_America-32)
conj_and(Ocean-18, India-20)
conj_and(Ocean-18, Asia-23)
conj_and(Ocean-18, America-32)
conj_and(Ocean-18, Africa-34)
```


nsubj(are_native-10, **Emu_novaehollandiae**-4)
prep_to(are_native-10, **Australia**-12)

nsubj(found-3, **jackdaws**-1)
conj_and(**North_west_India**-22, **Iran**-19)
conj_and(**North_west_India**-22, **Siberia**-24)
rcmod(**North_west_India**-22, inhabit-28)

nsubj(occurs-6, **Spotted Nutcracker**-5)
prep_in(occurs-6, **Europe**-8)
prep_in(occurs-6, **Asia**-10)
conj_and(**Europe**-8, **Asia**-10)
poss(**Nutcracker**-17, Clark-15)
prep_in(**Nutcracker**-17, **western_North_America**-21)

nsubj(are_birds-4, **Kiwi**-1)
prep_to(endemic-5, **New_zealand**-8)

Relation -located_in(Bird,Location)

nsubj(are_found-3, **Potoos**-1)
prep_in(are_found-3, Central-6)
prep_in(found-3, **South_American_country**-10)
prep_except(are_found-3, **Chile**-12)

nsubj(live-2, **Cranes**-1)
prep_on(live-2, continents-5)
conj_and(**Antarctica**-7, **South_America**-9)
prep_except(continents-5, **South_America**-10)

nsubj(found-6, **Barn_swallows**-1)
conj_and(**Australia**-11, **New_Zealand**-13)
prep_for(found-6, **New_Zealand**-14)
prep_for(found-6, **Madagascar**-16)
conj_and(**New_Zealand**-14, **Madagascar**-16)
conj_and(**New_Zealand**-14, regions-22)

nsubj(hunted-11, **Ostriches**-3)
prep_in(**Ostriches**-3, **Middle_East**-9)

nsubj(absent-3, **Swans**-1)
prep_from(absent-3, **Asia**-6)
prep_from(absent-3, **Central_America**-9)
conj_and(**Asia**-6, **Central_America**-9)
amod(**South_America**-13, northern-11)
prep_from(absent-3, **South_America**-13)
conj_and(**Asia**-6, **South_America**-13)
conj_and(**Asia**-6, **entirety_Africa**-16)

Relation: related(Bird,Bird)

nsubj(related-10, **Eurasian_Magpie**-6)
prep_to(related-10, **Eurasian_Jay**-14)
prep_to(than-15, **Green_Magpies**-22)
conj_and(**Blue**-19, **Green_Magpies**-22)

nsubj(are_group-5, **Treepies**-1)
prep_to(similar-9, **magpies**-11)

nsubj(share-2, **Ostriches**-1)
prep_with(share-2, **emus**-7)
prep_with(share-2, **kiwis**-9)
conj_and(**emus**-7, **kiwis**-9)
conj_and(**emus**-7, **ratites**-13)

nsubj(are_part-3, **Cassowaries**-1)
dobj(includes-11, **Emu**-13)
dobj(includes-11, **rheas**-15)
conj_and(**Emu**-13, **rheas**-15)
dobj(includes-11, **ostriches**-17)
conj_and(**Emu**-13, **ostriches**-17)
dobj(includes-11, **kiwis**-20)
conj_and(**Emu**-13, **kiwis**-20)
dobj(includes-11, **moas**-25)
conj_and(**Emu**-13, **moas**-25)
conj_and(**Emu**-13, **elephant_birds**-28)
conj_and(**moas**-25, **elephant_birds**-28)

nsubj(genus-5, **nutcrackers**-2)
prep_to(related-18, **jays**-21)
prep_to(related-18, **crows**-23)
conj_and(**jays**-21, **crows**-23)

nsubjpass(considered-11, **Kagu**-8)
prep_to(related-12, **adzebills**-16)
prep_from(**adzebills**-16, **New_Zealand**-19)
prep_from(**adzebills**-16, **Sunbittern**-22)
conj_and(**New_Zealand**-19, **Sunbittern**-22)

nsubj(is_relative-12, **Sunbittern**-7)
prep_of(is_relative-12, **Kagu**-15)

quantmod(**crows**-8, **as**-7)

```

prep_to(referred-5, crows-8)
nsubj(include-16, species_of_crows-11)
dobj(include-16, jackdaws-17)
dobj(include-16, rooks-19)
conj_and(jackdaws-17, rooks-19)

```

```

nsubj(constitute-4, Pigeons-1)
conj_and(Pigeons-1, doves-3)
nsubj(constitute-4, doves-3)

```

Relation: **¬related(Bird,Bird)**

```

nsubjpass(related-4, Parrots-1)
neg(related-4, not-3)
prep_to(related-4, owls-6)

```

```

nsubj(are-3, Humming_birds-2)
neg(are-3, not-4)
prep_as(family-8, magpies-10)

```

```

nsubj(is_bird-12, Southern_Cassowary-3)
conj_only(bird-12, ostrich-18)
conj_only(bird-12, emu-20)
conj_and(ostrich-18, emu-20)

```

```

nsubj(is_magpie-15, leucopterus-6)
cc(magpie-15, nor-16)
dobj(believed-21, jay-24)
dobj(believed-21, treepie-28)
conj_but(jay-24, treepie-28)

```

```

nsubjpass(known-8, bats-2)
prep_such_as(bats-2, lyra-6)
prep_on(known-8, Barn_Swallows-13)

```

```

nsubj(is_species-13, Kakapo-2)
appos(Kakapo-2, habroptila-5)
dobj(called-7, owl_parrot-9)

```

```

nsubj(take-12, birds_of_pre-2)
prep_such_as(birds_of_pre-2, Northern_Goshawks-9)
dobj(take-12, ducks-13)

```

Relation **has_characteristic(jj,Part)**

```

nsubj(is_distinctive-3, Ostrich-1)

```

```

amod(neck-11, long-10)
prep_with(distinctive-3, neck-11)
prep_with(distinctive-3, legs-13)
conj_and(neck-11, legs-13)
conj_and(neck-11, ability-16)

nsubj(have-2, Parrots-1)
amod(beak-6, curved-4)
amod(beak-6, red-5)
dobj(have-2, beak-6)

nsubj(is bill-11, distinctive_feature_of_hornbills-4)
amod(bill-11, heavy-10)
partmod(bill-11, supported-13)
amod(neck_muscles-17, powerful-15)
agent(supported-13, neck_muscles-17)
amod(vertebrae-24, fused-23)
agent(supported-13, vertebrae-24)
conj_and(muscles-17, vertebrae-24)

nsubj(is-5, plumage_of_hornbills-2)
prep_on(colors-21, bill-24)
amod(skin-31, bare-29)
amod(skin-31, colored-30)
prep_of(patch-27, skin-31)
prep_on(offset-18, face-34)
conj_or(face-34, wattles-36)

nsubj(has-6, male-2)
prep_in(male-2, plumage-5)
amod(head-9, chocolate-brown-8)
dobj(has-6, head-9)
amod(breast-12, white-11)
dobj(has-6, breast-12)
conj_and(head-9, breast-12)
prep_with(head-9, stripe-16)
prep_of(side-20, neck-23)

nsubj(is_bluish-4, bill-2)
nsubj(are_blue-grey-9, legs-7)

nsubj(have-12, all-11)
amod(bills-14, broad-13)
dobj(have-12, bills-14)
amod(tips-17, hooked-16)
prep_with(have-12, tips-17)
amod(wings-20, rounded-19)

```

```

prep_with(have-12, wings-20)
conj_and(tips-17, wings-20)
amod(legs-24, strong-23)
prep_with(have-12, legs-24)
conj_and(tips-17, legs-24)

nsubj(are_yellowish-green-6, breast-2)
conj_and(breast-2, flank-4)
nsubj(are-yellowish-green-6, flank-4)

nsubj(streaked-13, belly-2)
conj_and(belly-2, undertail-4)
nsubj(streaked-13, undertail-4)
conj_and(belly-2, neck-6)
nsubj(streaked-13, neck-6)
conj_and(belly-2, face-8)
nsubj(streaked-13, face-8)
rcmod(belly-2, yellowish-11)

nsubj(are_large-4, Kakapo_feet-2)
nsubj(scaly-6, Kakapo_feet-2)

amod(claws-3, pronounced-2)
nsubj(are_useful-6, claws-3)

nsubj(are_longer-5, Woodpecker_bills-2)
nsubj(sharper-7, Woodpecker_bills-2)
nsubj(stronger-9, Woodpecker_bills-2)
prep_than(longer-5, bills_of_piculets-12)

amod(tongues-4, long-2)
amod(tongues-4, sticky-3)
nsubj(possess-7, tongues-4)
nsubj(bristles-8, tongues-4)
nsubj(aid-10, tongues-4)
rcmod(tongues-4, possess-7)

nsubj(possess-7, Woodpeckers-1)
nsubj(possess-7, piculets-3)
nsubj(possess-7, wrynecks-5)
amod(feet-9, zygodactyl-8)
dobj(possess-7, feet-9)

amod(claws-6, strong-5)
prep_in_addition_to(have-10, claws-6)
conj_and(claws-6, feet-8)
prep_in_addition_to(have-10, feet-8)

```

```

nsubj(have-10, woodpeckers-9)
amod(legs-13, short-11)
amod(legs-13, strong-12)
doobj(have-10, legs-13)

poss(head-2, Its-1)
nsubj(are_black-9, head-2)
conj_and(head-2, neck-4)
nsubj(are_black-9, neck-4)
conj_and(head-2, breast-6)
nsubj(are_black-9, breast-6)
nsubj(are_white-28, belly-19)
conj_and(belly-19, scapulars-21)
nsubj(are_white-28, scapulars-21)
appos(belly-19, shoulder_feathers-24)
nsubj(are_glossed-34, wings-31)
amod(webs-46, white-44)
amod(webs-46, inner-45)
doobj(have-43, webs-46)
appos(webs-46, conspicuous-48)
nsubj(is_open-53, wing-51)

nsubj(are_black-6, legs-2)
conj_and(legs-2, bill-4)
nsubj(black-6, bill-4)

nsubj(include-5, features_of_parrots-2)
amod(bill-9, strong-7)
amod(bill-9, curved-8)
doobj(include-5, bill-9)
conj_and(bill-9, stance-13)
amod(legs-16, strong-15)
doobj(include-5, legs-16)
conj_and(bill-9, legs-16)
amod(feet-21, clawed-19)
amod(feet-21, zygodactyl-20)
doobj(include-5, feet-2)
conj_and(bill-9, feet-21)

nsubj(have-2, Parrots-1)
amod(wings-5, long-3)
amod(wings-5, broad-4)
doobj(have-2, wings-5)

nsubj(are_dense-12, feathers-2)
nsubj(are_silky-14, feathers-2)
prep_on(feathers-2, head-5)

```

```

prep_on(feathers-2, neck-7)
conj_and(head-5, neck-7)
prep_on(feathers-2, shoulders-9)
conj_and(head-5, shoulders-9)

```

Relation **-has_characteristic(jj,Part)**

```

nsubj(have-4, Cassowaries-1)
neg(have-4, not-3)
mod(bill-7, strong-6)
dobj(have-4, bill-7)

```

Taxonomic Relation

Relation **is_a(Bird, jj_Bird)**

```

nsubj(flightless_bird-5, Ostrich-1)
cop(flightless_bird-5, is-2)

```

```

nsubj(passering_birds-7, true_crows-3)
cop(passerine_birds-7, are-4)
nsubj(comprise-9, passerine_birds-7)
rmod(passerine_birds-7, comprise-9)

```

```

nsubj(passerine_birds-4, Magpies-1)
cop(passerine_birds-4, are-2)
prep_of(passerine_birds-4, crow_family-8)
appos(crow_family-8, Corvidae-10)

```

```

nsubj(aquatic_flightless_group_of_birds-4, Penguins-1)
cop(aquatic_flightless_group_of_birds-4, are-2)
partmod(aquatic_flightless_group_of_birds-9, living-10)

```

```

nsubj(large_passerine_family_of_bird_species -6, cotingas-2)
cop(large_passerine_family_of_bird_species -6, are-3)
partmod(large_passerine_family_of_bird_species-10, found-11)

```

```

conj_or(Pintail-2, Northern-4)
nsubj(occurring_duck-10, Northern_Pintail-5)
cop(occurring_duck-10, is-6)
nsubj(breeds-12, occurring_duck-10)
rmod(occurring_duck-10, breeds-12)

```

```

nsubj(run-9, Flightless_birds-2)
prep_such_as(Flightless_birds-2, ostrich-5)

```

```

appos(ostrich-5, emu-7)

nsubj(part-3, Cassowaries-1)
cop(part-3, are-2)
nsubj(includes-11, ratite_group-7)
rcmod(ratite_group-7, includes-11)
doobj(includes-11, Emu-13)
doobj(includes-11, rheas-15)
conj_and(Emu-13, rheas-15)
doobj(includes-11, ostriches-17)
conj_and(Emu-13, ostriches-17)
doobj(includes-11, kiwis-20)
conj_and(Emu-13, kiwis-20)
doobj(includes-11, moas-25)
conj_and(Emu-13, moas-25)
conj_and(Emu-13, elephant_birds-28)
conj_and(moas-25, elephant_birds-28)

```

Relation \neg is_a(Bird, jj_Bird)

```

nsubj(aquatic_bird-7, Magpie-2)
cop(aquatic_bird-7, is-3)
neg(aquatic_bird-7, not-4)

nsubj(magpie-6, Black_magpie-2)
cop(magpie-6, is-3)
advmod(magpie-6, neither-4)
conj_nor(magpie-6, jay-9)

```

Relation eat(Bird,Diet)

```

nsubj(consists-6, diet_of_ostrich-2)
prep_of(consists-6, plant_matter-9)
mark(eats-13, though-11)
nsubj(eats-13, ostrich-12)
advcl(consists-6, eats-13)
doobj(eats-13, insects-14)

```

```

nsubj(eats-2, Jackdaws-1)
nsubj(take-30, Jackdaws -1)
doobj(eats-2, insects-3)
doobj(eats-2, invertebrates-6)
conj_and(insects-3, invertebrates-6)
appos(insects-3, weed_seeds-9)

```


appos(**insects**-3, **grain**-11)
 conj_and(**seeds**-9, **grain**-11)
 appos(**insects**-3, **scraps_of_human_food**-13)
 prep_in(**scraps_of_human_food**-16, towns-18)
 appos(**insects**-3, **stranded_fish**-21)
 prep_on(**stranded_fish**-21, shore-24)
 conj_and(eats-2, take-30)
 dobj(take-30, **food**-31)
 rep_from(take-30, bird tables-34)

nsubj(consists-7, **diet_of_Pelican**-2)
 prep_of(consists-7, **fish**-9)
 nsubj(eat-14, they-12)
 conj_but(consists-7, eat-14)
 dobj(eat-14, **amphibians**-15)
 appos(**amphibians**-15, **crustaceans**-17)
 prep_on(smaller birds-24, occasions-21)
 conj_but(consists-7, birds-24)
 conj_and(eat-14, smaller birds-24)

nsubj(eat-3, **Most storks**-2)
 dobj(eat-3, **earthworms**-10)
 dobj(eat-3, **small birds**-14)
 conj_and(**earthworms**-10, birds-14)
 dobj(eat-3, mammals-16)
 conj_or(**earthworms**-10, mammals-16)

nsubj(eat-2, **Cranes**-1)
 dobj(sized-9, **small rodents**-11)
 dobj(sized-9, **fish**-13)
 conj_and(**rodents**-11, **fish**-13)
 dobj(sized-9, **amphibians**-15)
 conj_and(**rodents**-11, **amphibians**-15)
 dobj(sized-9, **insects**-18)
 conj_and(**rodents**-11, **insects**-18)
 prep_to(sized-9, **grain**-21)
 prep_to(sized-9, **berries**-23)
 conj_and(**grain**-21, **berries**-23)
 prep_to(sized-9, **plants**-26)
 conj_and(**grain**-21, **plants**-26)

nsubj(feed-2, **Doves**-1)
 prep_on(feed-2, **seeds**-4)
 prep_on(feed-2, **fruit**-6)
 conj_and(**seeds**-4, **fruit**-6)
 prep_on(feed-2, **plants**-8)

```

conj_and(seeds-4, plants-8)

nsubj(famous-4, They-1)
dobj(hunting-6, fish-9)
dobj(eating-8, fish-9)
dobj(catching-17, fish-18)
nsubj(take-23, other species-22)
dobj(take-23, crustaceans-24)
appos(crustaceans-24, frogs-26)
appos(crustaceans-24, other amphibians-29)
conj_and(frogs-26, amphibians-29)
appos(crustaceans-24, annelid worms-32)
appos(crustaceans-24, molluscs-34)
appos(crustaceans-24, insects-36)
appos(crustaceans-24, spiders-38)
appos(crustaceans-24, centipedes-40)
appos(crustaceans-24, reptiles-42)
pobj(including-44, snakes-45)
dobj(take-23, birds-49)
conj_and(crustaceans-24, birds-49)
conj_and(crustaceans-24, mammals-51)
conj_and(birds-49, mammals-51)

nsubj(consists-7, food of Haban Kukula-2)
prep_of(consists-7, grain-9)
prep_of(consists-7, weed seeds-12)
conj_and(grain-9, seeds-12)
prep_of(consists-7, berries-14)
conj_and(grain-9, berries-14)
prep_of(consists-7, various succulent leaves-18)
conj_and(grain-9, leaves-18)
conj_and(grain-9, buds-20)
conj_and(leaves-18, buds-20)
conj_and(grain-9, large proportion-25)

nsubj(consists-6, diet of Ostrich-2)
nsubj(are_carnivores-10, Most gulls-2)
appos(gulls-2, species-6)
nsubj(take-14, carnivores-10)
rcmod(carnivores-10, take-14)
dobj(take-14, food-16)
dobj(take-14, scavenge-18)
conj_or(food-16, scavenge-18)

```

Relation \neg eat(Bird,Diet)

nsubj(consume-4, **Gulls**-1)

neg(consume-4, not-3)

dobj(consume-4, **plant matter**-6)

nsubj(are accidental-8, **vegetable matter**-3)

partmod(matter-3, consumed-4)

agent(consumed-4, **Cranes**-6)

Domain - Sport

Relation: played(Method, Equipment)

```
nsubj(score-2, Players-1)
prepc_by(score-2, striking-5)
dobj(striking-5, shuttlecock-7)
poss(racquet-10, their-9)
prep_with(striking-5, racquet-10)
prep_over(passes-14, net-17)
conj_and(net-17, lands-19)
```

```
nsubj(is-6, object-2)
prepc_by(is-6, passing-14)
dobj(passing-14, ball-16)
prepc_by(is-6, shooting-18)
conj_and(passing-14, shooting-18)
dobj(shooting-18, it-19)
prep_into(shooting-18, goal-25)
```

```
nsubj(played-3, It-1)
xcomp(played-3, using-4)
conj_and(cue-6, snooker_balls-8)
dobj(using-4, snooker_balls-9)
```

```
nsubj(any-6, game_of_football-2)
partmod(degrees-21, kicking-23)
dobj(kicking-23, ball-25)
prep_with(ball-25, foot-28)
prep_in(kicking-23, attempt-31)
xcomp(kicking-23, score-33)
```

```
nsubj(sport-6, Golf-1)
xsubj(hit-24, players-11)
prep_of(types-18, clubs-20)
xcomp(attempt-22, hit-24)
dobj(hit-24, balls-25)
prep_into(hit-24, hole-28)
```

```
nsubj(family_of_sports-4, Hockey-1)
nsubj(play-11, teams-10)
xcomp(trying-16, maneuver-18)
dobj(maneuver-18, ball-20)
dobj(maneuver-18, puck-23)
conj_or(ball-20, puck-23)
```

```
prep_into(maneuver-18, goal-28)  
dobj(using-29, stick-32)
```

Relation: \neg played(Method, Equipment)

```
nsubj(played-4, Basket_ball-2)  
neg(played-4, not-5)  
agent(played-4, throwing-7)  
dobj(throwing-7, ball-9)  
cc(passing-11, but-10)  
conj_and(bouncing-13, shooting-15)
```

Appendix H

Relation Extraction Rules

Domain: Bird

Relation: located_in(Bird, Location)

$$\begin{aligned} \forall x \forall y (& \text{nsbj}(\text{VB}, x) \wedge \text{conj_and}(y, y) \wedge \neg \text{prep_from}(\text{VB}, y) \wedge \\ & \neg \text{prep_for}(\text{VB}, y) \wedge \neg \text{prep_except}((\text{NN} \vee y) \wedge \neg \text{negative}(\text{VB}) \wedge \neg \text{neg}(\text{VB}, \text{not}) \\ & \quad \longrightarrow \text{located_in}(x, y)) \quad 0.88 \\ \forall x \forall y (& \text{nsbj}(\text{VB}, x) \wedge \text{conj_and}(\text{VB}, y) \wedge \neg \text{prep_from}(\text{VB}, y) \wedge \\ & \neg \text{prep_for}(\text{VB}, y) \wedge \neg \text{prep_except}((\text{NN} \vee y) \wedge \neg \text{negative}(\text{VB}) \wedge \neg \text{neg}(\text{VB}, \text{not}) \\ & \quad \longrightarrow \text{located_in}(x, y)) \quad 0.8 \\ \forall x \forall y (& \text{nsbj}(\text{VB}, x) \wedge \text{prep_in}(\text{VB}, y) \wedge \neg \text{prep_except}((\text{NN} \vee y), y) \wedge \\ & \neg \text{neg}(\text{VB}, \text{not}) \wedge \neg \text{negative}(\text{VB}) \longrightarrow \text{located_in}(x, y)) \quad 1.75 \\ \forall x \forall y (& \text{nsbj}(\text{VB}, x) \wedge \text{prep_on}(\text{VB}, y) \wedge \neg \text{prep_except}((\text{NN} \vee y), y) \wedge \\ & \neg \text{neg}(\text{VB}, \text{not}) \wedge \neg \text{negative}(\text{VB}) \longrightarrow \text{located_in}(x, y)) \quad 0.95 \\ \forall x \forall y (& \text{nsbj}(\text{VB}, x) \wedge \text{prep_to}(\text{VB}, y) \wedge \neg \text{negative}(\text{VB}) \wedge \neg \text{neg}(\text{VB}, \text{not}) \\ & \quad \longrightarrow \text{located_in}(x, y)) \quad 1.87 \end{aligned}$$

Where $x \in \text{Bird}$, $y \in \text{location}$

Relation: related(Bird, Bird)

$$\begin{aligned} \forall x \forall y (& \text{nsbj}(\text{VB}, x) \wedge \text{conj_and}(\text{VB}, y) \wedge \neg \text{conj_only}(\text{NN}, y) \wedge \neg \text{negative}(\text{VB}) \\ & \quad \text{related}(x, y)) \quad 0.48 \\ \forall x \forall y (& \text{nsbj}(\text{VB}, x) \wedge \text{conj_and}((x, y) \wedge \neg \text{conj_only}(\text{NN}, y) \wedge \neg \text{negative}(\text{VB}) \\ & \quad \longrightarrow \text{related}(x, y)) \quad 0.5 \\ \forall x \forall y (& \text{nsbj}(\text{VB}, x) \wedge \text{prep_to}(\text{VB}, y) \wedge \neg \text{neg}(\text{VB}, \text{not}) \wedge \neg \text{negative}(\text{VB}) \\ & \quad \longrightarrow \text{related}(x, y)) \quad 0.76 \\ \forall x \forall y (& \text{nsbj}(\text{VB}, x) \wedge \text{dobj}(\text{VB}, y) \wedge \neg \text{cc}(x \vee y, \text{cc}) \wedge \neg \text{negative}(\text{VB}) \\ & \quad \longrightarrow \text{related}(x, y)) \quad 0.51 \\ \forall x \forall y (& \text{nsbj}(\text{VB}, x) \wedge \text{prep_of}(\text{VB}, y) \wedge \neg \text{negative}(\text{VB}) \\ & \quad \longrightarrow \text{related}(x, y)) \quad 0.46 \end{aligned}$$

Where $x, y \in \text{Bird}$,

Relation: eat(Bird, Diet)

$$\begin{aligned} \forall x \forall y (& \text{nsbj}(\text{VB}, x) \wedge \text{dobj}(\text{VB}, y) \wedge \neg \text{prep_except}((\text{NN} \vee y) \neg \text{negative}(\text{VB}) \wedge \\ & \neg \text{neg}(\text{VB}, \text{not}) \longrightarrow \text{eat}(x, y)) \quad 1.97 \\ \forall x \forall y (& \text{nsbj}(\text{VB}, x) \wedge \text{con_and}(y, y) \wedge \neg \text{negative}(\text{VB}) \wedge \neg \text{neg}(\text{VB}, \text{not}) \\ & \quad \longrightarrow \text{eat}(x, y)) \quad 0.67 \end{aligned}$$

$$\begin{aligned}
\forall x \forall y (nsubj(VB, x) \wedge prep_on(VB, y) \wedge \neg negative(VB) \wedge \neg neg(VB, not) &\longrightarrow eat(x, y)) \quad 1.74 \\
\forall x \forall y (nsubj(VB, x) \wedge xcomp(VB, y) \wedge \neg negative(VB) \wedge \neg neg(VB, not) &\longrightarrow eat(x, y)) \quad 1.21 \\
\forall x \forall y (nsubj(VB, y) \wedge prep_of(VB, y) \wedge \neg negative(VB) \wedge \neg neg(VB, not) &\longrightarrow eat(x, y)) \quad 0.7
\end{aligned}$$

Where $x \in \text{Bird}$, $y \in \text{Diet}$

Relation: nest_in(Bird, Nest)

$$\begin{aligned}
\forall x \forall y (nsubj(VB, x) \wedge prep_in(VB, y) \wedge \neg neg(VB, not) &\longrightarrow nest_in(x, y)) \quad 1.88 \\
\forall x \forall y (nsubj(VB, NN) \wedge prep_in(VB, y) \wedge \neg neg(VB, not) &\longrightarrow nest_in(x, y)) \quad 1.6 \\
\forall x \forall y (nsubj(VB, x) \wedge prep_on(VB, y) \wedge \neg neg(VB, not) &\longrightarrow nest_in(x, y)) \quad 1.21 \\
\forall x \forall y (nsubj(VB, x) \wedge dobj(VB, y) \wedge \neg neg(VB, not) &\longrightarrow nest_in(x, y)) \quad 0.74
\end{aligned}$$

Where $x \in \text{Bird}$, $y \in \text{Nest}$

Relation: has_characteristic((Bird, Bird_Part)

$$\begin{aligned}
\forall x (nsubj((VB, x) \wedge amod(x, jj) \wedge \neg neg(VB, not) &\longrightarrow has_characteristic(jj, x)) \\
\forall x \forall y (nsubj(VB, y) \wedge prep_with(VB, x) \wedge \neg neg(VB, not) &\longrightarrow has_characteristics(y, x)) \quad 2.1 \\
\forall y \forall z (nsubj(VB, y) \wedge prep_with(NN, z) \wedge \neg neg(VB, not) &\longrightarrow has_characteristics(y, z)) \quad 0.8 \\
\forall x \forall y (nsubj(VB, y) \wedge dobj(VB, x) \wedge \neg neg(VB, not) &\longrightarrow has_characteristics(y, x)) \quad 2.5
\end{aligned}$$

Relation: has_characteristic (Bird_Part, jj)

$$\begin{aligned}
\forall y (nsubj(jj, y) \wedge cop(jj, VB) \wedge \neg neg(VB, not) &\longrightarrow has_characteristic(y, jj)) \quad 1.97 \\
\forall y (nsubj(jj, y) \wedge conj_and(jj, jj) &\longrightarrow has_characteristic(y, jj)) \quad 1.2 \\
\forall y \forall z (nsubj(z, y) \wedge cop(z, VB) \wedge \neg neg(VB, not) &\longrightarrow has_characteristic(y, z)) \quad 0.87
\end{aligned}$$

Where $x \in \text{Bird_Part}$, $y \in \text{Bird}$, $z \in \text{Colour}$

Relation: lay_eggs(Bird, Egg_number)

$$\begin{aligned}
\forall x \forall y (nsubj(VB, x) \wedge dobj(VB, y) \wedge \neg neg(VB, not) &\longrightarrow lay_eggs(x, y)) \quad 2.67 \\
\forall x \forall y (nsubj(VB, x) \wedge prep_to(VB, y) \wedge \neg neg(VB, not) &\longrightarrow lay_eggs(x, y)) \quad 0.86
\end{aligned}$$

$$\begin{aligned} \forall x \forall y (nsubj(VB, x) \wedge nmod_by(verb, Bird) \wedge \neg neg(VB, not) &\longrightarrow lay_eggs(x, y)) & 2.4 \\ \forall x \forall y (nsubj(VB, x) \wedge nmod_for(VB, x) \wedge \neg neg(VB, not) &\longrightarrow lay_eggs(x, y)) & 0.5 \end{aligned}$$

Where $x \in Bird, y \in N$

Relation: has_length(Bird, Length)

$$\begin{aligned} \forall x \forall y (nsubj(VB, x) \wedge cop(y, VB) &\longrightarrow has_length(x, y)) & 1.97 \\ \forall x \forall y (nsubj(VB, x) \wedge prep_to(VB, y) &\longrightarrow has_length(x, y)) & 1.21 \\ \forall x \forall y (nsubj(VB, x) \wedge nmod_nmod(VB, y) &\longrightarrow has_length(x, y)) & 2.03 \\ \forall x \forall y (nsubj(VB, x) \wedge nmod_from(VB, y) &\longrightarrow has_length(x, y)) & 0.98 \\ \forall x \forall y (nsubj(VB, x) \wedge dobj(VB, y) &\longrightarrow has_length(x, y)) & 0.67 \\ \forall x \forall y (nsubj(VB, x) \wedge prep_at(VB, y) &\longrightarrow has_length(x, y)) & 0.37 \end{aligned}$$

Where $x \in Bird, y \in Length$

Relation: has_weight(Bird, Weight)

$$\begin{aligned} \forall x \forall y (nsubj(VB, x) \wedge prep_to(VB, y) &\longrightarrow has_weight(x, y)) & 0.72 \\ \forall x \forall y (nsubj(VB, x) \wedge dobj(VB, y) &\longrightarrow has_weight(x, y)) & 0.60 \\ \forall x \forall y (nsubj(VB, x) \wedge prep_from(VB, y) &\longrightarrow has_weight(x, y)) & 0.62 \\ \forall x \forall y (nsubj(VB, x) \wedge prep_at(VB, y) &\longrightarrow has_length(x, y)) & 0.38 \end{aligned}$$

Where $x \in Bird, y \in Weight$

Taxonomic Relation

Relation is_a(Bird, Super_Bird)

$$\begin{aligned} \forall x \forall y (nsubj(y, x) \wedge cop(y, VB) \wedge \neg neg(x, not) \wedge \neg conj_nor(x, x) &\longrightarrow is_a(x, y)) & 2.1 \\ \forall x \forall y (nsubj(VB, y) \wedge prep_such_as(y, x) &\longrightarrow is_a(x, y)) & 2.0 \\ \forall x \forall y (nsubj(VB, y) \wedge prep_such_as(y, x) \wedge appos(x, x) &\longrightarrow is_a(x, y)) & 1.98 \end{aligned}$$

Where $x \in Bird, y \in Super_Bird$

Domain: Sport

Relation: played(Method, Equipment)

$$\forall x \forall y (\text{nsbj}((\text{VB}, x) \wedge \text{dobj}(x, y) \wedge \neg \text{neg}(x, \text{not}) \wedge \neg \text{negative}(\text{VB})) \longrightarrow \text{played}(x, y)) \quad 1.36$$

$$\forall x \forall y (\text{nsbj}((\text{VB}, x) \wedge \text{conj}(y, y) \wedge \neg \text{neg}(x, \text{not}) \wedge \neg \text{negative}(\text{VB})) \longrightarrow \text{played}(x, y)) \quad 0.7$$

Where $x \in \text{Method}$, $y \in \text{Equipment}$

Relation: played_with(Sport, Equipment)

$$\forall x \forall y (\text{nsbj}((\text{VB}, \text{NN}) \wedge \text{dobj}(\text{VB}, y) \wedge \neg \text{neg}(y, \text{not}) \wedge \neg \text{negative}(\text{VB})) \longrightarrow \text{played_with}(x, y)) \quad 0.96$$

$$\forall x \forall y (\text{nsbj}((\text{VB}, \text{NN}) \wedge \text{prep_with}(\text{VB}, y) \wedge \neg \text{neg}(y, \text{not}) \wedge \neg \text{negative}(\text{VB})) \longrightarrow \text{played_with}(x, y)) \quad 1.18$$

Where $x \in \text{Sport}$, $y \in \text{Equipment}$

Relation: made_of(Equipment, Material)

$$\forall x \forall y (\text{nsbj}(\text{VB}, x) \wedge \text{dobj}(\text{VB}, y) \wedge \neg \text{neg}(\text{VB}, \text{not}) \longrightarrow \text{made_of}(x, y)) \quad 1.01$$

$$\forall x \forall y (\text{nsbj}(\text{VB}, x) \wedge \text{prep_of}(\text{VB}, y) \wedge \neg \text{neg}(\text{VB}, \text{not}) \longrightarrow \text{made_of}(x, y)) \quad 1.65$$

Where $x \in \text{Sport}$, $y \in \text{Material}$

Relation: has_player(Sport, Player_No)

$$\forall x \forall y (\text{nsbj}(\text{VB}, x) \wedge \text{nummod}(\text{NN}, y) \wedge \neg \text{neg}(\text{VB}, \text{not}) \longrightarrow \text{has_player}(x, y)) \quad 1.13$$

$$\forall x \forall y (\text{nsbj}(\text{VB}, y) \wedge \text{prep_in}(\text{VB}, x) \wedge \neg \text{neg}(\text{VB}, \text{not}) \longrightarrow \text{has_player}(x, y)) \quad 0.66$$

$$\forall x \forall y (\text{nsbj}(\text{VB}, x) \wedge \text{nummod}(y, y) \wedge \neg \text{neg}(\text{VB}, \text{not}) \longrightarrow \text{has_player}(x, y)) \quad 1.11$$

$$\forall x \forall y (\text{nsbj}(\text{VB}, x) \wedge \text{conj_or}(y, y) \wedge \neg \text{neg}(\text{VB}, \text{not}) \longrightarrow \text{has_player}(x, y)) \quad 0.53$$

$$\forall x \forall y (\text{nsbj}(\text{VB}, \text{NN}) \wedge \text{nummod}(\text{NN}, y) \wedge \neg \text{neg}(\text{VB}, \text{not}) \longrightarrow \text{has_player}(x, y)) \quad 1.07$$

Where $x \in \text{Sport}$, $y \in \mathbb{N}$

Relation: has_weight(Equipment, Weight)

$$\forall x \forall y (\text{nsbj}(\text{VB}, x) \wedge \text{dobj}(\text{NN}, y) \wedge \neg \text{neg}(\text{VB}, \text{not}) \longrightarrow \text{has_weight}(x, y)) \quad 0.82$$

$$\forall x \forall y (\text{nsbj}(\text{VB}, x) \wedge \text{prep_of}(\text{NN}, y) \longrightarrow \text{has_weight}(x, y)) \quad 1.91$$

$$\forall x \forall y (\text{nsbj}(\text{VB}, x) \wedge \text{cop}(y, \text{VB}) \longrightarrow \text{has_weight}(x, y)) \text{ 0.89}$$

Where $x \in \text{Equipment}$, $y \in \text{Weight}$

Relation: has_length(Equipment, Length)

$$\forall x \forall y (\text{nsbj}(y, x) \longrightarrow \text{has_length}(x, y)) \text{ 0.26}$$

$$\forall x \forall y (\text{nsbj}(\text{VB}, x) \wedge \text{prep_of}(\text{NN}, y) \longrightarrow \text{has_length}(x, y)) \text{ 1.78}$$

$$\forall x \forall y (\text{nsbj}(\text{VB}, x) \wedge \text{cop}(y, \text{VB}) \longrightarrow \text{has_length}(x, y)) \text{ 0.85}$$

Where $x \in \text{Equipment}$, $y \in \text{Length}$

Relation: has_width(Equipment, Width/Diameter)

$$\forall x \forall y (\text{nsbj}(\text{VB}, x) \wedge \text{prep_of}(\text{NN}, y) \longrightarrow \text{has_width}(x, y)) \text{ 1.80}$$

$$\forall x \forall y (\text{nsbj}(\text{VB}, x) \wedge \text{cop}(y, \text{VB}) \longrightarrow \text{has_width}(x, y)) \text{ 0.83}$$

Where $x \in \text{Equipment}$, $y \in \text{Width}$, $y \in \text{Diameter}$

Relation: played_in(Sport, Location)

$$\forall x \forall y (\text{nsbj}(\text{VB}, x) \wedge \text{prep_in}(\text{VB}, y) \wedge \neg \text{prep_except}((\text{NN} \vee y), y) \wedge \neg \text{neg}(\text{VB}, \text{not}) \wedge \neg \text{negative}(\text{VB}) \longrightarrow \text{located_in}(x, y)) \text{ 2.1}$$

$$\forall x \forall y (\text{nsbj}(\text{VB}, x) \wedge \text{prep_of}(\text{VB}, y) \wedge \neg \text{prep_except}((\text{NN} \vee y), y) \wedge \neg \text{neg}(\text{VB}, \text{not}) \wedge \neg \text{negative}(\text{VB}) \longrightarrow \text{located_in}(x, y)) \text{ 0.91}$$

Where $x \in \text{Sport}$, $y \in \text{location}$

Relation: is_a(Sport, Super_sport)

$$\forall x \forall y (\text{nsbj}(y, x) \wedge \text{cop}(y, \text{VB}) \wedge \neg \text{neg}(x, \text{not}) \longrightarrow \text{is_a}(x, y))$$

Where $x \in \text{Sport}$, $y \in \text{Super_Sport} \vee \text{Equipment}$

Reuters - 21578 corpus

Category acq

$\forall x \forall y ((\text{nsubj}(\text{VB}, x) \wedge \text{prep_for}(\text{VB}, y) \wedge \neg \text{negative}(\text{VB}) \wedge \neg \text{neg}(\text{VB}, \text{not})) \longrightarrow \text{Acquire}(x, y))$ 1.56
 $\forall x \forall y ((\text{nsubj}(\text{VB}, x) \wedge \text{prep_for}(x, y) \wedge \neg \text{negative}(\text{VB}) \wedge \neg \text{neg}(\text{VB}, \text{not})) \longrightarrow \text{Acquire}(x, y))$ 1.54
 $\forall x \forall y ((\text{nsubj}(\text{VB}, x) \wedge \text{prep_for}(\text{NN}, y) \wedge \neg \text{negative}(\text{VB}) \wedge \neg \text{neg}(\text{VB}, \text{not})) \longrightarrow \text{Acquire}(x, y))$ 1.43
 $\forall x \forall y ((\text{nsubj}(\text{VB}, x) \wedge \text{prep_at}(\text{VB}, y) \wedge \neg \text{negative}(\text{VB}) \wedge \neg \text{neg}(\text{VB}, \text{not})) \longrightarrow \text{Acquire}(x, y))$ 0.32
Where $x \in \text{Organization}$, $y \in \text{share_price}$ $\text{VB} \in \{\text{acquire}, \text{buy}\}$

$\forall x \forall y (\text{nsubj}(\text{VB}, x) \wedge \text{dobj}(\text{VB}, y) \wedge \neg \text{negative}(\text{VB}) \wedge \neg \text{neg}(\text{VB}, \text{not})) \longrightarrow \text{Acquire}(x, y))$ 1.28
 $\forall x \forall y (\text{nsubj}(\text{VB}, x) \wedge \text{prep_for}(\text{VB}, y) \wedge \neg \text{negative}(\text{VB}) \wedge \neg \text{neg}(\text{VB}, \text{not})) \longrightarrow \text{Acquire}(x, y))$ 0.46
Where $x \in \text{Organization}$, $y \in \text{no_of_shares}$ $\text{VB} \in \{\text{acquire}, \text{buy}\}$

$\forall x \forall y ((\text{nsubj}(\text{VB}, x) \wedge \text{prep_for}(x, y) \wedge \neg \text{negative}(\text{VB}) \wedge \neg \text{neg}(\text{VB}, \text{not})) \longrightarrow \text{Sell}(x, y))$ 1.97
 $\forall x \forall y ((\text{nsubj}(\text{VB}, x) \wedge \text{prep_for}(\text{NN}, y) \wedge \neg \text{negative}(\text{VB}) \wedge \neg \text{neg}(\text{VB}, \text{not})) \longrightarrow \text{Sell}(x, y))$ 1.95
 $\forall x \forall y ((\text{nsubj}(\text{VB}, x) \wedge \text{prep_of}(y, \text{NN}) \wedge \neg \text{negative}(\text{VB}) \wedge \neg \text{neg}(\text{VB}, \text{not})) \longrightarrow \text{Sell}(x, y))$ 1.04
Where $x \in \text{Organization}$, $y \in \text{share_price}$ $\text{VB} \in \{\text{sell}, \text{sold}, \text{completed}\}$

$\forall x \forall y (\text{nsubj}(\text{VB}, x) \wedge \text{dobj}(\text{VB}, y) \wedge \neg \text{negative}(\text{VB}) \wedge \neg \text{neg}(\text{VB}, \text{not})) \longrightarrow \text{Sell}(x, y)$ 1.78
 $\forall x \forall y (\text{nsubj}(\text{VB}, x) \wedge \text{prep_of}(y, \text{NN}) \wedge \neg \text{negative}(\text{VB}) \wedge \neg \text{neg}(\text{VB}, \text{not})) \longrightarrow \text{Sell}(x, y)$ 0.91
Where $x \in \text{Organization}$, $y \in \text{no_of_shares}$, $\text{VB} \in \{\text{sell}, \text{sold}, \text{completed}\}$

$\forall x \forall y (\text{nsubj}(\text{VB}, x) \wedge \text{dobj}(\text{VB}, y) \wedge \neg \text{negative}(\text{VB}) \wedge \neg \text{neg}(\text{VB}, \text{not})) \longrightarrow \text{Sell}(x, y)$ 1.81
 $\forall x \forall y (\text{nsubj}(\text{VB}, x) \wedge \text{prep_of}(\text{NN}, y) \wedge \neg \text{negative}(\text{VB}) \wedge \neg \text{neg}(\text{VB}, \text{not})) \longrightarrow \text{Sell}(x, y)$ 0/67
Where $x \in \text{Organization}$, $y \in \text{Service}$, $y \in \text{Product}$, $\text{VB} \in \{\text{sell}, \text{sold}\}$

$$\begin{aligned} \forall x \forall y ((\text{nsubj}(\text{VB}, x) \wedge \text{prep_to}(\text{NN}, y) \wedge \neg \text{negative}(\text{VB}) \wedge \neg \text{neg}(\text{VB}, \text{not})) \\ \longrightarrow \text{Sell_to}(x, y)) \quad 0.74 \\ \forall x \forall y ((\text{nsubj}(\text{VB}, x) \wedge \text{prep_to}(x, y) \wedge \neg \text{negative}(\text{VB}) \wedge \neg \text{neg}(\text{VB}, \text{not})) \\ \longrightarrow \text{Sell_to}(x, y)) \quad 0.77 \end{aligned}$$

Where $x, y \in \text{Organization}$

$$\begin{aligned} \forall x \forall y (\text{conj_and}(x, y) \wedge \text{dobj}(\text{VB}, \text{NN}) \longrightarrow \text{merge_with}(x, y) \quad 0.43 \\ \forall x \forall y (\text{nsubj}(\text{VB}, x) \wedge \text{prep_with}(\text{VB}, y) \longrightarrow \text{merge_with}(x, y) \quad 1.01 \\ \text{Where } x, y \in \text{Organization}, \text{VB} \in \{\text{become}, \text{merge}\} \end{aligned}$$

$$\begin{aligned} \forall x \forall y (\text{nsubj}(\text{VB}, x) \wedge \text{dobj}(\text{VB}, y) \wedge \neg \text{negative}(\text{VB}) \wedge \neg \text{neg}(\text{VB}, \text{not})) \\ \longrightarrow \text{Earn_profit}(x, y) \quad 1.25 \\ \forall x \forall y (\text{nsubj}(\text{VB}, x) \wedge \text{prep_of}(\text{NN}, y) \wedge \neg \text{negative}(\text{VB}) \wedge \neg \text{neg}(\text{VB}, \text{not})) \\ \longrightarrow \text{Earn_profit}(x, y) \quad 1.75 \\ \text{Where } x \in \text{Organization}, y \in \text{profit}, \text{VB} \in \{\text{reported}, \text{announced}, \text{posted}\} \end{aligned}$$

$$\begin{aligned} \forall x \forall y (\text{nsubj}(\text{VB}, x) \wedge \text{dobj}(\text{VB}, y) \wedge \neg \text{negative}(\text{VB}) \wedge \neg \text{neg}(\text{VB}, \text{not})) \\ \longrightarrow \text{provide}(x, y) \quad 1.13 \\ \forall x \forall y (\text{nsubj}(\text{VB}, y) \wedge \text{prep_of}(\text{NN}, x) \wedge \neg \text{neg}(\text{VB}, \text{not})) \\ \longrightarrow \text{provide}(x, y) \quad 0.94 \\ \text{Where } x \in \text{Organization}, y \in \text{Product/Service}, \text{VB} \in \{\text{produce}, \text{provide}, \\ \text{is_production}\} \end{aligned}$$

Category bop

$$\begin{aligned} \forall x \forall y (\text{prep_for}(\text{NN}, x) \wedge \text{nummod}(\text{NN}, y) \longrightarrow \text{has_current_account_deficit}(x, y)) \\ 0.78 \\ \forall x \forall y (\text{prep_in}(\text{NN}, x) \wedge \text{nummod}(\text{NN}, y) \longrightarrow \text{has_current_account_deficit}(x, y)) \\ 0.77 \end{aligned}$$

Where $x \in \text{Period}$, $y \in \text{current_account_deficit}$

$$\begin{aligned} \forall x \forall y \forall z (\text{nsubj}(\text{VB}, x) \wedge \text{nummod}(z, y) \longrightarrow \text{has_current_account_deficit}(x, y)) \\ 2.02 \\ \forall x \forall y \forall z (\text{nsubj}(y, \text{NN}) \wedge \text{nummod}(\text{NN}, x) \longrightarrow \text{has_current_account_deficit}(x, y)) \\ 2.0 \end{aligned}$$

Where $x \in \text{Period}$, $y \in \text{current_account_deficit}$, $z \in \text{Currency}$

$$\begin{aligned} \forall x \forall y (\text{nsubj}(\text{VB}, x) \wedge \text{prep_to}(\text{VB}, y) \wedge \neg \text{negative}(\text{VB}) \wedge \neg \text{neg}(\text{VB}, \text{not})) \\ \longrightarrow \text{has_current_account_deficit}(x, y) \\) \\ 1.06 \\ \forall x \forall y (\text{nsubj}(\text{VB}, x) \wedge \text{prep_from}(\text{VB}, y) \wedge \neg \text{negative}(\text{VB}) \wedge \neg \text{neg}(\text{VB}, \text{not})) \\ \longrightarrow \text{had_current_account_deficit}(x, y)) \\ 0.91 \end{aligned}$$

Where $x \in \text{Country}$, $y \in \text{current_account_deficit}$

$\forall x \forall y (\text{nsbj}(\text{VB}, x) \wedge \text{dobj}(\text{VB}, y) \wedge \neg \text{negative}(\text{VB}) \wedge \neg \text{neg}(\text{VB}, \text{not})$
 $\longrightarrow \text{has_current_account_surplus}(x, y))$ 2.7

$\forall x \forall y (\text{nsbj}(\text{VB}, x) \wedge \text{prep_of}(\text{NN}, y) \wedge \neg \text{negative}(\text{VB}) \wedge \neg \text{neg}(\text{VB}, \text{not})$
 $\longrightarrow \text{has_current_account_surplus}(x, y))$ 1.48

Where $x \in \text{Period}$, $y \in \text{current_account_surplus}$

$\forall x \forall y \forall z (\text{nsbj}(\text{VB}, x) \wedge \text{nummod}(z, y) \longrightarrow \text{has_trade_surplus}(x, y))$ 1.78

$\forall x \forall y (\text{nsbj}(\text{VB}, x) \wedge \text{nummod}(\text{NN}, y) \longrightarrow \text{has_trade_surplus}(x, y))$ 1.53

$\forall x \forall y (\text{nummod}(\text{NN}, y) \longrightarrow \text{has_trade_surplus}(x, y))$

Where $x \in \text{Country}$, $y \in \text{trade_surplus}$, $z \in \text{Currency}$

$\forall x \forall y (\text{nsbj}(y, \text{NN}) \wedge \text{prep_for}(\text{NN}, x) \longrightarrow \text{import}(x, y))$ 0.75

$\forall x \forall y \forall z (\text{nummod}(\text{NN}, y) \wedge \text{nummod}(z, x) \longrightarrow \text{import}(x, y))$ 1.04

Where $x \in \text{Period}$, $y \in \text{Imports}$, $z \in \text{Currency}$

$\forall x \forall y (\text{nummod}(\text{NN}, x) \wedge \text{nummod}(z, y) \longrightarrow \text{has_burrowing}(x, y))$ 0.43

$\forall x \forall y (\text{prep_in}(\text{NN}, x) \wedge \text{prep_to}(\text{VB}, y) \wedge \neg \text{neg}(\text{VB}, \text{not}) \longrightarrow \text{has_burrowing}(x, y))$
0.21

Where $x \in \text{Period}$, $y \in \text{burrowing}$,

Category dlr

$\forall x \forall y (\text{nsbj}(\text{VB}, x) \wedge \text{prep_at}(\text{VB}, y) \longrightarrow \text{has_dollar_value}(x, y))$ 1.82

$\forall x \forall y (\text{nummod}(x, y) \longrightarrow \text{has_dollar_value}(x, y))$ 1.31

Where $x \in \text{Currency}$, $y \in \text{dollar_value}$ $\text{VB} \in \{\text{trading, put}\}$

$\forall z \forall y (\text{nummod}(\text{NN}, y) \wedge \text{nummod}(x, z) \longrightarrow \text{has_dollar_value_in}(y, z))$ 0.97

$\forall z \forall y (\text{nummod}(x, z) \wedge \text{prep_in}(z, y) \longrightarrow \text{has_dollar_value_in}(y, z))$ 0.99

Where $x \in \text{Currency}$, $y \in \text{Year}$, $z \in \text{dollar_value}$

$\forall x \forall y (\text{nsbj}(x, x) \wedge \text{nummod}(x, y) \longrightarrow \text{has_value}(x, y))$ 0.37

Where $x \in \text{Currency}$, $y \in \text{Currency_value}$

$\forall x \forall y (\text{prep_wth}(\text{NN}, x) \wedge \text{prep_by}(\text{VB}, y) \longrightarrow \text{rise_import_with}(x, y))$ 0.72

$\forall x \forall y (\text{prep_wth}(\text{NN}, x) \wedge \text{prep_for}(\text{VB}, y) \longrightarrow \text{rise_import_with}(x, y))$ 0.70

Where $x \in \text{Country}$, $y \in \text{Rise_in_Import}$

$\forall x \forall y (\text{nsbj}(\text{VB}, x) \wedge \text{prep_by}(x, y) \longrightarrow \text{rise_currency_rate}(x, y))$ 0.42

Where $x \in \text{Currency}$, $y \in \text{Currency_value}$

$\forall x \forall y (\text{nsbj_xsubj}(\text{VB}, x) \wedge \text{dobj}(\text{VB}, y) \longrightarrow \text{post_dollar_rate_surplus}(x, y))$ 1.05

$\forall x \forall y (\text{dobj}(\text{VB}, x) \wedge \text{dep}(x, y) \longrightarrow \text{post_dollar_rate_surplus}(x, y))$ 0.69
 Where $x \in \text{Country}$, $y \in \text{Dollar_Rate_Surplus}$

Category earn

$\forall x (\text{prep_from}(\text{NN}, x) \wedge \text{prep_to}(\text{VB}, x) \longrightarrow \text{increase_stock}(x, x))$ 0.63
 $\forall x \forall y (\text{nsbj}(\text{VB}, y) \wedge \text{prep_to}(\text{VB}, x) \longrightarrow \text{increase_stock}(y, x))$ 1.37

$\forall y \forall z (\text{nsbj}(\text{VB}, y) \wedge \text{prep_below}(\text{VB}, z) \longrightarrow \text{has_profit_below}(y, z))$ 1.03
 $\forall z \forall p (\text{nmod_poss}(p, \text{PRP\$}) \wedge \text{prep_below}(\text{VB}, z) \longrightarrow \text{has_profit_below}(p, z))$ 0.7

Where $x \in \text{Stock_amt}$, $y \in \text{Organization}$, $z \in \text{Profit}$, $p \in \text{Period}$

$\forall x \forall y (\text{nsbj}(\text{VB}, \text{NN}) \wedge \text{nunmod}(\text{VB}, x) \longrightarrow \text{has_expenditure}(y, x))$ 1.85
 $\forall x \forall y (\text{nsbj}(\text{VB}, y) \wedge \text{dobj}(\text{VB}, x) \longrightarrow \text{has_expenditure}(y, x))$ 1.88
 Where $x \in \text{Expenditure}$, $y \in \text{Organization}$,

$\forall x \forall y (\text{nsbj}(\text{VB}, \text{NN}) \wedge \text{prep_to}(\text{VB}, x) \longrightarrow \text{has_profit}(y, x))$ 0.56
 $\forall x \forall y (\text{nsbj}(\text{VB}, y) \wedge \text{dobj}(\text{VB}, x) \longrightarrow \text{has_profit}(y, x))$ 1.25
 Where $x \in \text{Profit}$, $y \in \text{Organization}$,

$\forall x \forall y (\text{nsbj}(\text{VB}, y) \wedge \text{prep_for}(\text{VB}, x) \longrightarrow \text{has_sales}(y, x))$ 1.57
 $\forall x \forall y (\text{nsbj}(\text{VB}, y) \wedge \text{prep_of}(\text{NN}, x) \longrightarrow \text{has_sales}(y, x))$ 1.61
 $\forall z \forall y (\text{nsbj}(\text{VB}, y) \wedge \text{dobj}(\text{VB}, z) \longrightarrow \text{has_sales}(y, z))$ 1.34
 Where $x \in \text{Sales}$, $y \in \text{Organization}$, $z \in \text{Service_unit}$

$\forall z \forall y (\text{nsbj}(\text{VB}, y) \wedge \text{dobj}(\text{VB}, z) \longrightarrow \text{earn}(y, z))$ 0.98
 $\forall z \forall y (\text{nsbj}(\text{VB}, y) \wedge \text{prep_of}(\text{NN}, z) \longrightarrow \text{earn}(y, z))$ 1.06
 Where $y \in \text{Organization}$, $z \in \text{Income}$

$\forall x \forall y (\text{nsbj}(\text{VB}, y) \wedge \text{prep_of}(\text{NN}, x) \longrightarrow \text{declare_dividend}(y, x))$ 0.83
 Where $y \in \text{Organization}$, $x \in \text{dividend}$

$\forall x \forall y (\text{nsbj}(\text{VB}, y) \wedge \text{nmod_tmod}(\text{jj}, x) \longrightarrow \text{pay_dividend}(y, x))$ 0.77
 $\forall x \forall y (\text{nsbj}(\text{VB}, y) \wedge \text{dep}(\text{jj}, x) \longrightarrow \text{pay_dividend}(y, x))$ 0.48
 Where $y \in \text{Organization}$, $x \in \text{Date}$

$\forall x \forall y (\text{nsbj}(\text{VB}, y) \wedge \text{dobj}(\text{VB}, x) \longrightarrow \text{produce}(y, x))$ 1.77
 $\forall x \forall y (\text{nsbj}(\text{VB}, y) \wedge \text{prep_for}(\text{VB}, x) \longrightarrow \text{produce}(y, x))$ 1.85
 Where $y \in \text{Organization}$, $x \in \text{Product}$

Category jobs

$\forall x \forall y (nsubj(VB, y) \wedge prep_to(VB, x) \longrightarrow has_unemployment_rate(y, x))$ 0.88
 $\forall x \forall y (nsubj(VB, y) \wedge prep_from(VB, x) \longrightarrow has_unemployment_rate(y, x))$ 0.86
 $\forall x \forall y (nsubj(VB, y) \wedge dobj(VB, x) \longrightarrow has_unemployment_rate(y, x))$ 1.01
 $\forall x \forall y (nsubj(VB, y) \wedge prep_at(VB, x) \longrightarrow has_unemployment_rate(y, x))$ 1.11
 Where $y \in Country, x \in Rate$

$\forall x \forall y (prep_to(VB, x) \wedge prep_in(VB, y) \longrightarrow has_unemployment_rate(y, x))$ 0.65
 $\forall x \forall y (prep_at(VB, x) \wedge prep_in(VB, y) \longrightarrow has_unemployment_rate(y, x))$ 0.54
 $\forall x \forall y (nsubj(VB, x) \wedge prep_in(VB, y) \longrightarrow has_unemployment_rate(y, x))$ 0.95
 Where $y \in Period, x \in Rate$

$\forall x \forall y (nsubj(VB, y) \wedge prep_at(VB, x) \longrightarrow has_femaleunemployment_rate(y, x))$ 0.81
 $\forall x \forall y (nsubj(VB, y) \wedge dobj(VB, x) \longrightarrow has_femaleunemployment_rate(y, x))$ 1.06
 Where $y \in Country, x \in Rate$

$\forall x \forall y (nsubj(VB, y) \wedge prep_at(VB, x) \longrightarrow has_maleunemployment_rate(y, x))$ 0.72
 $\forall x \forall y (nsubj(VB, y) \wedge dobj(VB, x) \longrightarrow has_maleunemployment_rate(y, x))$ 1.22
 Where $y \in Country, x \in Rate$

$\forall x \forall y (prep_in(VB, y) \wedge prep_to(VB, x) \longrightarrow has_employment_rate(y, x))$ 0.66
 $\forall x \forall y (nsubj(VB, y) \wedge dobj(VB, x) \longrightarrow has_employment_rate(y, x))$ 0.71
 Where $y \in Industry, x \in Rate$

$\forall x \forall y (nsubj(VB, y) \wedge dobj(VB, x) \longrightarrow rise_employment_rate(y, x))$ 0.93
 $\forall x \forall y (nsubj(VB, x) \wedge prep_in(VB, y) \longrightarrow rise_employment_rate(y, x))$ 1.16
 Where $y \in Employment, x \in Rate$

Categoryv ships

$\forall x \forall y (nsubj(VB, x) \wedge prep_in(VB, y) \longrightarrow transfer_charters_to(y, x))$ 0.56
 $\forall x \forall y (nsubj(VB, x) \wedge prep_to(VB, y) \longrightarrow transfer_charters_to(y, x))$ 0.54
 Where $x, y \in Shippingline$

$\forall x \forall y (perp_of(NN, x) \wedge prep_at(NN, y) \longrightarrow has_capacity(y, x))$ 0.76
 $\forall x \forall y (nsubj(VB, x) \wedge dobj(VB, x) \wedge \neg neg(VB, not) \longrightarrow has_capacity(y, x))$ 1.02

Where $y \in Port, x \in Capacity$

$\forall x \forall y (nsubj(VB, x) \wedge nsubj(VB, y) \longrightarrow halt_at(y, x))$ 0.55
 $\forall x \forall y (nsubj(VB, y) \wedge prep_to(VB, x) \longrightarrow halt_at(y, x))$ 0.61
 Where $y \in Ship, x \in Port \quad VB \in \{was_closed, is_closed, ran, hit, grounded\}$

$\forall x \forall y (\text{nummod}(\text{NN}, x) \wedge \text{prep_in}(\text{NN}, y) \longrightarrow \text{is_halt}(x, y)) \ 0.75$

$\forall x \forall y (\text{nummod}(\text{NN}, x) \wedge \text{dobj}(\text{VB}, y) \longrightarrow \text{is_halt}(x, y)) \ 0.70$

Where $y \in \text{Location}$, $x \in \text{No_Ships}$ $\text{VB} \in \{\text{wait}, \text{halt}\}$

$\forall x \forall y (\text{nsubj}(\text{VB}, x) \wedge \text{dobj}(\text{VB}, y) \longrightarrow \text{carry}(x, y)) \ 1.74$

$\forall x \forall y (\text{nsubj}(\text{VB}, x) \wedge \text{prep_to}(\text{VB}, y) \longrightarrow \text{carry}(x, y)) \ 1.01$

$\forall x \forall y (\text{nsubj}(\text{VB}, x) \wedge \text{conj_and}(y, y) \longrightarrow \text{carry}(x, y)) \ 0.99$

Where $y \in \text{Goods}$, $x \in \text{Ship}$, $\text{VB} \in \{\text{deliver}, \text{transport}, \text{carry}\}$

$\forall x \forall y (\text{nsubj}(\text{VB}, x) \wedge \text{prep_to}(\text{VB}, y) \longrightarrow \text{lease_ships}(x, y)) \ 2.01$

$\forall x \forall y (\text{nsubj}(\text{VB}, x) \wedge \text{prep_from}(\text{VB}, y) \longrightarrow \text{lease_ships}(y, x)) \ 1.96$

Where $x, y \in \text{Shippingline}$,

$\forall x \forall y (\text{nsubj}(\text{VB}, x) \wedge \text{prep_with}(\text{VB}, y) \longrightarrow \text{has_agreement}(x, y)) \ 0.68$

Where $y \in \text{Shippingline}$, $x \in \text{Country}$

$\forall x \forall y (\text{nsubj}(\text{VB}, x) \wedge \text{prep_with}(\text{VB}, y) \longrightarrow \text{has_agreement}(x, y)) \ 0.62$

Where $x \in \text{Shippingline}$, $y \in \text{Country}$

$\forall x \forall y (\text{prep_from}(\text{VB}, x) \wedge \text{dobj}(\text{VB}, y) \longrightarrow \text{carry}(x, y)) \ 0.55$

$\forall x \forall y (\text{nsubj}(\text{VB}, x) \wedge \text{prep_with}(\text{VB}, y) \longrightarrow \text{carry}(x, y)) \ 1.01$

Where $y \in \text{No_people}$, $x \in \text{Ship}$,

$\forall x \forall y (\text{nsmmod}(\text{NN}, x) \wedge \text{prep_from}(\text{VB}, y) \longrightarrow \text{rescue_from}(x, y)) \ 1.01$

$\forall x \forall y (\text{nsubj}(\text{VB}, y) \wedge \text{nummod}(\text{NN}, x) \longrightarrow \text{rescue_from}(x, y)) \ 1.18$

Where $x \in \text{No_people}$, $y \in \text{Ship}$,

$\forall x \forall y (\text{nsubj}(\text{VB}, x) \wedge \text{prep_from}(\text{VB}, y) \longrightarrow \text{sale_from}(x, y)) \ 1.74$

$\forall x \forall y (\text{nsubj}(\text{VB}, x) \wedge \text{dobj}(\text{VB}, y) \longrightarrow \text{sale_from}(x, y)) \ 1.51$

Where $y \in \text{Port}$, $x \in \text{Ship}$, $\text{VB} \in \{\text{left}, \text{was_off}\}$

$\forall y (\text{nsubj}(\text{VB}, x) \longrightarrow \text{is_closed}(y)) \ 2.3$

$\forall y (\text{nsubj}(\text{VB}, \text{NN}) \wedge \text{dobj}(\text{VB}, y) \longrightarrow \text{is_closed}(y)) \ 1.05$

Where $y \in \text{Port}$, $\text{VB} \in \{\text{closed}, \text{was_closed}\}$

Category trade

$\forall x \forall y (\text{prep_to}(\text{VB}, x) \wedge \text{nummod}(\text{NN}, y) \longrightarrow \text{has_inflation_rate}(x, y)) \ 0.97$

$\forall x \forall y (\text{nsubj}(\text{VB}, x) \wedge \text{dobj}(\text{VB}, y) \longrightarrow \text{has_inflation_rate}(x, y)) \ 1.85$

Where $x \in \text{Country}$, $y \in \text{Inflation_rate}$,

$\forall x \forall y (\text{nsubj}(\text{VB}, x) \wedge \text{prep_to}(\text{VB}, y) \longrightarrow \text{has_trade_surplus}(x, y)) \ 1.27$

$\forall x \forall y (\text{nsubj}(\text{VB}, x) \wedge \text{dobj}(\text{VB}, y) \longrightarrow \text{has_trade_surplus}(x, y)) \ 1.99$

Where $x \in \text{Country}$, $y \in \text{Trade_surplus}$, $\text{VB} \in \{\text{plunged, jumped, widened, was}\}$

$\forall x \forall y (\text{prep_to}(\text{VB}, y) \wedge \text{prep_jn}(\text{VB}, x) \longrightarrow \text{has_surplus_in}(y, x)) 0.62$

$\forall x \forall y (\text{dojb}(\text{VB}, y) \wedge \text{prep_for}(\text{NN}, x) \longrightarrow \text{has_surplus_in}(y, x)) 0.61$

Where $x \in \text{Period}$, $y \in \text{Trade_surplus}$

$\forall x \forall y (\text{prep_to}(\text{VB}, x) \wedge \text{prep_from}(\text{VB}, y) \longrightarrow \text{rise_surplus}(x, y)) 0.83$

Where $x, y \in \text{Trade_surplus}$

$\forall x \forall y (\text{nsubj}(\text{VB}, x) \wedge \text{prep_to}(\text{VB}, y) \longrightarrow \text{has_debt}(x, y)) 0.87$

$\forall x \forall y (\text{nsubj}(\text{VB}, x) \wedge \text{dojb}(\text{VB}, y) \longrightarrow \text{has_debt}(x, y)) 1.01$

Where $x \in \text{Country}$, $y \in \text{Debt}$, $\text{VB} \in \{\text{cover, had, has}\}$

$\forall x \forall y (\text{dojb}(\text{VB}, x) \wedge \text{prep_to}(\text{VB}, y) \longrightarrow \text{rise_export}(x, y)) 0.69$

$\forall x \forall y (\text{prep_to}(\text{VB}, y) \wedge \text{prep_by}(\text{VB}, x) \longrightarrow \text{rise_export}(x, y)) 0.66$

Where $x \in \text{Percent}$, $y \in \text{Export}$,

$\forall x \forall y (\text{dojb}(\text{VB}, x) \wedge \text{prep_to}(\text{VB}, y) \longrightarrow \text{fall_import}(x, y)) 0.97$

$\forall x \forall y (\text{prep_to}(\text{VB}, y) \wedge \text{prep_by}(\text{VB}, x) \longrightarrow \text{fall_import}(x, y)) 0.78$

Where $x \in \text{Percent}$, $y \in \text{import}$,

Appendix I

Relations and Relation Instances found

Domain: Bird

Relations between entities Bird and Location

Relation: located(Bird, Location)

Positive verbs - {live, are_native, occur, are_found, colonise, establish, are_restricted, is_dominant}

Negative verbs – {absent, extinct}

Relation Instances found for the relation located_in(Bird, Location)	New Relations established between bird and location
(Albatross, Southern Ocean) (Petrel, Southern Ocean) (Eagle, Eurasia) (Flamingo, America) (Macaw, Mexico) (Macaw, Caribbean) (Hornbill, Africa) (Hornbill, Asia) (Cassowary, New Guinea) (Kakapo, New Zealand) (Falcon, Europe) (Falcon, North America) (Grebe, South America) (Pelican, France) (Auk, California) (Cuckoo, North America) (Cuckoo, South America) (Cuckoo, Canada) (Eagle, Eurasia)	farmed_in(Bird, Location) (Ostrich, Sweden) (Ostrich, Finland) is_national_bird(Bird, Location) (Peacock, India) (Barn swallow, Estonia) (Junglefowl, Sri Lanka) endangered_in(Bird, Location) (Cassowary, Australia) worshipped_in (Eagle, Peru)

(Eagle, Africa)	
(Gannet, Southern Africa)	
(Gannet, Australia)	
(Gannet, Newzealand)	
(Spoonbill, Europe)	
(Spoonbill, Asia)	
(Spoonbill, Japan)	
(Vulture, North America)	
(Vulture, South America)	
(Vulture, Africa)	
(Vulture, Asia)	
(Shoebill, east_Africa)	

Relations between the entities Bird and Bird

Relation: related(Bird, Bird)

Positive verbs - {relate, share}

Negative verbs – {unrelated, called, is_similar, associate, prey_for, is_called}

Relation Instances found for the relation related(Bird, Bird)	New Relations established between bird and Bird
(Frigatebird, Pelican)	associated_with
(Falcon, Pelican)	(Swift, Hummingbird)
(Grebe, Loon)	(Darter, Stork)
(Grebe, Flamingo)	(Darter, Herons)
(Shoebill, Hammerkop)	
(Stork, Herons)	is_similar_to
(Stork, Spoonbill)	(Treepie, Magpie)
(Turcos, Cuckoo)	(Auk, Penguin)
(Swift, Humming bird)	
(Gannet, Booby)	prey_for

	(Duck, Goshawks) (Bat, Barnswallow) is_called (Kakapo, Owl parrot)
--	---

Relations between the entities Bird and characteristic of bird body part

Relation: has_characteristic(Bird, Bird_Part)

Relation Instances found for the relation has_characteristic(Bird, jj_Part)	New Relations established between bird and characteristic of bird body part
(Albatross, large_Bill) (Albatross, dark_upper_Wing) (Frigatebird, large_Wing) (Falcon, long_Wing) (Grebe, narrow_Wing) (Pelican, long_Beak) (Pelican, large_Throat) (Stork, long_Neck) (Stork, stout_Bill) (Turaco, long_Tail) (Nightjar, long_Wings) (Owl, hawk-like_Beak) (Owl, wide_Face) (Swift, short_forked_Tail) (Swift, swept_back_Wings) (Vulture, bald_Head) (Vulture, devoid_of_normal_feathers) (Eagle, heavy_Head) (Hawk, long_Tail)	Since two entities concerned in this relation are bound by an adjective not by a verb new relations cannot be established .

(Spoonbill, spatulate_bill)	
Relation Instances found for the relation has_characteristic(Bird_Part, jj)	
Heron (bill, long) (bill, harpoon-like) (wings, broad) (wings, long) Shoebill (plumage, blue-grey) (wings, broad) Grebe (feet, long) Kakapo (eyes, dark_brown) (flank, yellowish-green) Cuckoo (feet, zygodactyl) Hammerkop (plumage, drab_brown) (tail, short) (wings, big) Bateleur (tail, small)	

Relations between entities Bird and Diet

Relation: eat(Bird, diet)

Positive verbs - {take, eat, consume, consist_of, feed_on, prey_on}

Relation Instances found for the relation eat(Bird, diet)	New Relations established between Bird and diet
(Albatross, cephalopods) (Albatross, fish) (Albatross, crustaceans) (Albatross, offal) (Flamingo, brine_shrimp) (Flamingo, blue-green_algae) (Macaw, seeds) (Macaw, nuts) (Macaw, fruits) (Macaw, leaves) (Macaw, flowers) (Macaw, stems) (Hornbill, fruit) (Hornbill, insects) (Hornbill, small_animals) (Cassowary, fruit) (Cassowary, shoots) (Kakapo, plants) (Kakapo, seeds) (Grebe, fish) (Grebe, freshwater_insects) (Grebe, own_feathers) (Pelican, fish) (Cuckoo, insect) (Cuckoo, insect_larvae)	

(Vulture, dead_animal)	
(Loon, fish)	
(Hammerkop, fish)	
(Hammerkop, amphibians)	
(Heron, aquatic_animals)	

Relations between entities Bird and Nest

Relation: nest_in(Bird, Nest)

Positive verbs - {nest_in, make, use}

Relation Instances found for the relation Nest_in(Bird, Nest)	New Relations established between Bird and Nest
(Hornbill, nests of woodpeckers) (Pelican, Trees) (Pelican, ground) (Cuckoo, trees) (Cuckoo, bushes) (Spoonbill, trees) (Spoonbill, reed_bed) (Hammerkop, fork_of_tree) (Hammerkop, bank) (Hammerkop, cliff) (Heron, ground)	

Relations between entities Bird and Length

Relation: has_length(Bird, Length)

Relation Instances found for the relation Has_length(Bird, Nest)	New Relations established between Bird and Length
(Hornbill, 30 cm) (Kakapo, 58cm) (Falcon, 65 cm) (Pelican, 1.06m) (Loon, 66 cm) (Loon, 91 cm) (Shoebill, 100 cm) (Shoebill, 140 cm) (Heron, 25 cm)	

Relations between entities Bird and Weight

Relation: has_weight(Bird, Weight)

Relation Instances found for the relation has_weight(Bird, Weight)	New Relations established between Bird and Weight
(Flamingo, 7.7 or 5.5 pounds) (Cassowary, 58.5 kg) (Kakapo, 2 kg) (Grebe, 120 g) (Pelican, 2.75kg) (Auk, 85g) (Auk, 1kg) (Cuckoo, 17g) (Cuckoo, 630g) (Loon, 2.2 kg)	

(Loon, 7.6 kg)	
(Shoebill, 5.6 kg)	
(Hammerkop, 470 g)	

Relations between entities Bird and Egg_number

Relation: lay_eggs(Bird, Egg_number)

Positive verbs - {lay, consists_of}

Relation Instances found for the relation lay_eggs(Bird, Egg_number)	New Relations established between Bird and Egg_number
(Albatross, 1) (Hornbil, 6) (Cassowary, 3 to 8) (Kakapo, 4) (Pelican, 2) (Auk, 1) (Gannet, 1) (Spoonbill, 3) (Loon, 1) (Loon, 2) (Hammerkop, 3) (Hammerkop, 7) (Heron, 3) (Heron, 7) (Booby, 1)	

Taxonomic Relation

Relation: is_a(Bird, jj_bird)

Relation Instances for the relation is_a(Bird, jj_bird)

(Albatross, Seabird)

(Darter, tropical_Waterbird)

(Frigatebird, family_of_Seabird)

(Ibis, long_legged_Wading_bird)

(Petrel, tube_nosed_Seabird)

(Spoonbill, group_of_large_long-legged_Wading_bird)

(Cuckoo, medium_sized_Bird)

(Hoatzin, species_of_Tropical_bird)

(Pelican, genus_of_large_Water_bird)

(Plover, distributed_group_of_Wading_bird)

(Swift, family_of_Aerial_bird)

Domain: Sport

Relations between entities Method, Equipment

Relation: played(Method, Equipment)

Positive verbs – {play, score}

Relation Instances found for the relation Play_with(Method, Equipment)	New Relations established between Method and Equipment
<p>Bandy</p> <p>(direct, ball)</p> <p>(propel, ball)</p> <p>(passing, ball)</p> <p>(touching, ball)</p> <p>Discus throw</p> <p>(throw, disc)</p> <p>Pato</p> <p>(throwing, ball)</p> <p>Lacrosse</p> <p>(using, small_rubber_ball)</p> <p>(shooting, ball)</p> <p>(using, stick)</p> <p>(catch, ball)</p> <p>Polo</p> <p>(driving, wooden_ball)</p> <p>Tejo</p> <p>(throwing, metal_plate)</p>	

(throwing, disc)	
------------------	--

Relations between entities Sport, Equipment

Relation: play_with(Sport, Equipment)

Positive verbs – {play, throw, use}

Relation Instances found for the relation played(Sport, Equipment)	New Relations established between Sport and Equipment
(Bandy, ball) (Bandy, stick) (Bandy, skate) (Discus throw, disc) (Pato, Ball) (Lacrosse, ball) (Lacrosse, stick) (Polo, Ball) (Polo, Mallet) (Tejo, plate) (Tejo, disc)	

Relations between entities Equipment, Material

Relation: made_of(Equipment,Material)

Positive verbs – {is_made, is_used}

Relation Instances found for the relation Made_of(Equipment, Material)	New Relations established between Equipment and Material
Bandy (stick, wood)	Polo comprise(mallet, cane_sharft)

Discus throw (disc, plastic) (disc, metal) (disc, fiberglass) (disc, rubber) Pato (ball, leather) Lacrosse (stick, aluminum) (stick, wood) Polo (ball, high_impact_plastic) Tejo (disc, metal)	
---	--

Relations between entities Sport, Player_number
Relation: has_player(Sport, Player_number)
Positive verbs – {played, consists_of, has}

Relation Instances found for the relation has_player(Sport, Player_number)	New Relations established between Sport and Player_number
(Bandy, 11) (Pato, 4) (Lacrosse, 10)	

Relations between entities Equipment, Length and Width/Diameter

Relations: has_length(Equipment, length), has_width(Equipment, Width/Diameter)

Relation Instances found for the relation has_length(Equipment, Length) has_width(Equipment, Width/Diameter)	New Relations established between Equipment, Length and Width/Diameter
has_length(Equipment, Length) Bandy (stick, 127 cm) Lacrosse (stick, 40 inches) (stick, 42 inches) (stick, 35 inches) (stick, 43.25 inches) has_width(Equipment, Width/Diameter) Bandy (stick, 7 cm) Discus Throw (disc, 22cm) (disc, 18 cm) Pato (ball, 40 cm) Lacrosse (stick, 12 inches) Polo (ball, 3 inches)	

Relations between entities Equipment, Weight

Relation: has_weight(Equipment, Weight)

Relation Instances found for the relation Has_weight(Equipment, Weight)	New Relations established between Equipment and Weight
<p>Discus throw</p> <p>(disc, 2 kg)</p> <p>(disc, 1 kg)</p> <p>Pato</p> <p>(ball, 1050 g)</p> <p>(ball, 1250 g)</p> <p>Polo</p> <p>(ball, 3.5 ounces)</p> <p>(ball, 4.5 ounces)</p>	<p>Polo</p> <p>has_minimum_weight(ball, 170g)</p>

Relations between entities Sportt, Location

Relation: played_in(Sport, Location)

Positive verbs – {is_played, is_national_sport, is_developed, was_started}

Negative verbs – {is_banned}

Relation Instances found for the relation Played_in(Sport, Location)	New Relations established between Sport and Location
<p>(Bandy, Sweden)</p> <p>(Bandy, Russia)</p> <p>(Pato, Argentina)</p> <p>(Polo,Brazil)</p> <p>(Polo, America)</p>	<p>directed_from(Polo, India)</p> <p>is_held(Russia, Bandy)</p>

(Polo, Chile)	
(Polo, Mexico)	
(Polo, Singapore)	
(Polo, Malaysia)	
(Tejo, Colombia)	

Relation is_a(Sport, Super_sport)

(Bandy, team_winter_sport)
(Discus Throw, track_and_field)
(Lacrosse, team_sport)
(Polo, horse_back_mounted_team_sport)
(Tejo, throwing_sport)

Appendix J

Semantic Patterns for the entities in the corpus Reuters - 21578

Category acq

Entity	Pattern	Example
Share_price	<CD> MONETORY UNIT per share	0.125 dlrs per share
No_of_shares	<CD> [mln] [JJ] shares sold/sell <CD> <CD> ORG [JJ] shares	20000 shares 10.1 mln shares 33 mln ordinary shares sold 110,000 32,800 Robeson common shares
Product	ORG make(s) <NN> ORG, <NN> and <NN> group Manufacture <NN> Supplier of <NN> [and] <NN> <NN> firm ORG	Computer terminal makes computer generated labels Nobel Industrier, an arms and chemicals group Manufacture cooling systems Supplier of enhancement products and disc drive subsystems diary equipment firm Alfa
Service	involved in <NN> offer <NN> [and] <NN> service	Involved in application of fertilizers Offer lawn and garden care service
Profit	Profits [VB] <IN CD> [mln] [MONETORY UNIT] [VB] revenues of <CD> [mln] [MONETORY UNIT]	Profits may be below 2.4 mln Dlrs Had revenues of 8.4 mln dlrs

	[VB] profits of <CD> [mln] [MONETORY UNIT]	Profits of 283,000 dlrs
Purchase_price	Bought/sold [No_of_shares] [ORD] [JJ] shares [IN] [DATE] for <CD> [mln] MONETORY UNIT Sold [its] subsidiaries engaged in SERVICE for <CD> [mln] MONETORY UNIT Bought stake for <CD> [mln] MONETORY UNIT	bought 362,700 Wrather common shares between Feb 13 and 24 for 6.6 mln dlrs sold its subsidiaries engaged in pipeline and terminal operations for 12.2 mln dlrs. bought the stake for 2.1 mln dlrs

Category bop

Entity	Pattern	Example
current_account_deficit	current account deficit for YEAR [of] <CD> Deficit of <CD> [mln] Current account deficit for [quarter] [ended] DATE narrowed to <CD> [mln] MONETORY UNIT from <CD> [mln] <CD> [mln] [crowns] deficit Adjusted <CD> MONETORY UNIT [in] [PERIOD] from deficit of <CD> MONETORY UNIT <CD> MONETORY UNIT from YEAR <CD> MONETORY UNIT Deficit VB [IN] [N] to <CD> MONETORY UNIT from MONETORY UNIT	current account deficit for 1986/87 of 14.75 billion. deficit of 334 mln. current account deficit for the quarter ended December 31, 1986 narrowed to 567 mln dlrs from 738 mln 100 mln crowns deficit adjusted 2.27 billion dlrs in the fourth quarter from a deficit of 1.94 billion dlrs 8.81 billion dlrs from 1985's 584 mln dlrs deficit deficit grew to 5.04 billion dlrs from 4.14 billion dlrs

	current account deficit VBD <CD> [mln] MONETORY UNIT current account VB in deficit by <CD> [mln] MONETORY UNIT	current account deficit was 912 mln dlrs current account was in deficit by 760 mln stg
Current_ac count_surpl us	current account surplus VB <CD> MONETORY UNIT	current account surplus was 4.65 billion dlrs
Merchant_s urplus	<CD> [mln] MONETORY UNIT surplus for merchandise trade trade surplus VB to <CD> [mln] MONETORY UNIT trade balance VB <CD> surplus [trade] [balance] [VB] [YEAR] surplus of <CD> surplus on merchandise trade to <CD> [mln] from [surplus] [of] <CD> [mln] Surplus [IN] [NN] trade VB <CD> MONETORY UNIT	182 mln dlr surplus for merchandise trade trade surplus narrowed to 110 mln dlrs in February from 525 mln trade balance was 2.3 billion surplus trade balance showed a 1986 surplus of 33.2 billion surplus on merchandise trade to 46 mln from a surplus of 33 mln surplus on oil trade was 4.0 billion stg
Export	Exports VB <CD> Exports VB to <CD> MONETORY UNIT	Exports were 2.837 billion exports rose to 2.87 billion dlrs
Import	Imports [for] [PERIOD] VB <CD> Imports VB to <CD> [mln] from <CD> [mln] Imports in [NN] VB [IN] <CD> MONETORY UNIT Imports [IN] [COUNTRY] VB <CD>	Imports for the December 1986 quarter were 55 billion imports fell to 858 mln from 895 mln imports in the quarter were up 2.7 billion dlrs imports from Canada fell 300

	MONETARY UNIT	mln dlrs
Gdp_growth	gdp growth of <CD> pct	gdp growth of 2.5 pct

Category dlr

Entity	Pattern	Example
Agreement	<NN> pact	New York Plaza pact
Economic_growth	economic growth [NN] [IN] <CD> pct <CD> pct growth [VB] Growth rate <VB> [for] [COUNTRY] to <CD> pct from <CD> pct	economic growth downwards to 2.5 pct 2.8 pct growth forecast Growth rate forecasts for West Germany to 2.2 pct from 3.2 pct

Category earn

Entity	Pattern	Example
Capital_stock	capital stock from <CD> [mln] to <CD> [mln] [shares]	capital stock from five mln to 25 mln shares.
Profit_margin	Profits VB below <CD> [mln] CURRENCY	profits may be below the 2.4 mln dlrs
Profit	Revenues VB to <CD> [mln] CURRENCY [from] [<CD>] [mln] [CURRENCY] Profits VB [IN] [YEAR] to <CD> [mln] CURRENCY gain [IN] [NN] of <CD> [mln] CURRENCY	Revenues rose to 291.8 mln dlrs from 151.1 mln dlrs profits rose in 1986 to 120 mln stg gain of 2.9 mln dlrs revenues of 8,157,864 dlrs

	Revenues IN <CD> [mln] CURRENCY Profit of <CD> [mln] CURRENCY <CD> [mln] profit	profit of 104 mln dlrs 155 mln profit
loss	Loss of <CD> [mln] CURRENCY	loss of 1.8 mln dlrs
dividend	Dividend of <CD> cents per share Dividend of <NN for NN> Dividend [IN] [NN] of about <CD> [mln] CURRENCY	dividend of three cents per share dividend of one class A share for each two class A shares dividend to North American Coal of about 10 mln dlrs
income	Income [IN] [NN] VB [IN] [VB] [NN] <CD> [mln] CURRENCY Earned <CD> [mln] CURRENCY Income of [CD] [CURRENCY] [NN] [or] <CD> [mln] CURRENCY Earnings IN <CD> [mln] CURRENCY VB <CD> [mln] CURRENCY [IN] [NN] income.	income for the first quarter is expected to be about 10.4 mln dlrs earned 23.1 mln dlrs Income of 10 cts a share or 330,000 dlrs earnings of 10.8 mln dlrs generated 5.9 mln dlrs in net income

Category jobs

Entity	Pattern	Example
Unemployment_rate	unemployment rate VB [IN] [NN] <CD> pct	unemployment rate rose to a record 3.0 pct
	unemployment VB [up] [from] <CD> pct	Unemployment was up from 2.8 pct
	unemployment VB <CD> [mln] people	unemployment totalled 1.82 mln people
	<CD> pct unemployment rate	3.0 pct unemployment rate
	unemployment rate VB [IN] [VB] [IN] [VB] [IN] about <CD> pct	unemployment rate is expected to continue to climb to about 3.5

	<CD> VB [IN] jobless <CD> VB unemployed jobless VB [IN] <CD>	pct 170,000 increase in jobless 173,000 people were unemployed jobless stood at 508,392
Male_unemployment_rate	male unemployment [IN] [PERIOD] VB [IN] <CD> pct male unemployment [IN] <CD> pct	Male unemployment in January remained at 2.9 pct Male unemployment of 3.1 pct
female_unemployment_rate	Female unemployment [IN] [PERIOD] VB [IN] <CD> pct	Female unemployment in January remained at 3.0 pct
Employment_rate	employment VB <CD>	employment rose 337,000

Category ship

Entity	Pattern	Example
port_capacity	<CD> [mln] tonnes [NN] capacity [PORT\	20 mln tonne a year capacity Tianjin port
Berth_capacity	berths [IN] [JJ] capacity of <CD> [mln] tonnes	berths with an annual capacity of 6.28 mln tonnes
Ship_cost	<CD> [mln] CURRENCY SHIP	320 mln dlr polar icebreaker
service	<NN> service	South American service
export	<NN>[and] [<NN>] exports	coffee and tea exports
import	<NN>[and] [<NN>] imports	oil and fertilizer imports
No_people	<CD> people	540 people
demand	want <NN VB>	want pay rises
No_ship_employee	<CD> SHIP_EMPLOYEE	40,000 seafarers

Category trade

Entity	Pattern	Example
Inflation	<CD> pct inflation [rate]	250 pct inflation rate.
debt	<CD> [billion] CURRENCY [foreign] debt	109 billion dlr foreign debt.
trade_surplus	trade surplus [IN] [JJ] [PERIOD] VB [IN] <CD> [mln] CURRENCY TRADE_SURPLUS from <CD> [billion]>	trade surplus plunged to 211 mln dlrs 3.58 billion dlrs from 1.94 billion
export_change	exports VB <CD> pct	exports rose 14.6 pct
export	EXPORT_CHANGE from [NN] [NN] [IN] [JJ] [PERIOD] to <CD> [billion]>	14.6 pct from a year earlier in the first 20 days of February to 10.91 billion
import_change	imports VB <CD> pct	imports fell 3.2 pct
import	IMPORT_CHANGE to <CD> [billion]>	3.2 pct to 7.33 billion
currency_rate	CURRENCY/CURRENCY rates VB <CD>	dollar/yen rates were 152.32
current_account_surpluses	current account surplus <VB> <CD> [billion] dlrs	current account surplus was 4.65 billion dlrs