AN APPROACH TOWARDS SELF-SUPERVISED CLASSIFICATION USING CYC

Kino High Coursey, B.A. C.S.

Thesis Prepared for the Degree of

MASTER OF SCIENCE

UNIVERSITY OF NORTH TEXAS

December 2006

APPROVED:

Rada Mihalcea, Major Professor
Paul Tarau, Co-Major Professor
Larry Lefkowitz, Committee Member
Armin R. Mikler, Departmental Program Coordinator
Krishna Kavi, Chair of the Department of Computer Science and Engineering
Oscar Garcia, Dean of the College of Engineering
Sandra L. Terrell, Dean of the Robert B. Toulouse School of Graduate Studies

Coursey, Kino High, An Approach Towards Self-Supervised Classification Using Cyc. Master of Science (Computer Science), December 2006, 133 pp., 4 tables, 22 figures, 41 references.

Due to the long duration required to perform manual knowledge entry by human knowledge engineers it is desirable to find methods to automatically acquire knowledge about the world by accessing online information. In this work I examine using the Cyc ontology to guide the creation of Naïve Bayes classifiers to provide knowledge about items described in Wikipedia articles. Given an initial set of Wikipedia articles the system uses the ontology to create positive and negative training sets for the classifiers in each category. The order in which classifiers are generated and used to test articles is also guided by the ontology. The research conducted shows that a system can be created that utilizes statistical text classification methods to extract information from an ad-hoc generated information source like Wikipedia for use in a formal semantic ontology like Cyc. Benefits and limitations of the system are discussed along with future work.

ACKNOWLEDGEMENTS

I want to thank my advisors Dr. Rada Mihalcea and Dr. Paul Tarau for their support and encouragement. They have been extremely helpful in clarifying topics that were once vague.

I want to thank Susan Pirzchalski for years of unconditional support and for providing both stimulating conversation and life.

I want to thank all the friends that have that have made learning fun, and for sharing both common and different dreams.

Also, I want to thank all those whose lives have served as both positive and negative examples.

And finally, I want to thank my family, both living and dead, who have both made it possible for me to be who I am and to be here today.

TABLE OF CONTENTS

Page

LIST OF TABLES

Page

LIST OF FIGURES

Page

LIST OF ACRONYMS

BFS	Breadth first search
CycL	Cyc logic level language
DAG	Directed acyclic graph
FN	False negative (articles mistakenly identified as not belonging to a class)
FP	False positive (articles mistakenly identified as belonging to a class)
KB	Knowledge base
KE	Text Knowledge entry text
SubL	Cyc’s Lisp-level language
TN	True negative (articles correctly identified as not belonging to a class)
TP	True positive (articles correctly identified as belonging to a class)

CHAPTER 1

INTRODUCTION

1.1 Research Problem and Motivation

Acquiring knowledge from reading online data sources has been a long-term goal of Artificial Intelligence and Natural Language processing. Manual entry of knowledge requires a great deal of time and training. Finding a way to extract descriptions about real world objects and events continues to be an important area of research.

The Cyc project is an example of a broad coverage ontology developed over many years to provide knowledge-based services to a wide range of application. Since its development additional resources have become available for free over the Internet, one of them being the volunteer-edited Wikipedia.

Methods exist to create text-based classifiers—systems that, given a set of positive and negative examples, can classify a new text as exhibiting that concept or not. The classic example is text-base spam classification. Just as a human can specify what is messages are spam or not to train a recognizer, given an initial set of specific articles Cyc can specify what articles are examples of a more generic concept or not to also train a recognizer.

Cyc's concepts are arranged in an ontology, a network of terms and their relationships. Given a set of easy to find mappings based on specific objects (like "Eiffel Tower" or "Jimmy Carter") one can use Cyc to collect positive and negative examples of text about more abstract concepts, like "Landmark" and "Leader." Once created, a classifier can

be used to identify Wikipedia articles and reference Web pages for association with concepts in the ontology.

A paraphrase of what I have termed "Lenat's bootstrap hypothesis" (by Cycorp founder Dr. Douglas Lenat) is that once a system like Cyc reaches a certain level/scale it can help in its own development and start using natural language processing to augment its knowledge base. This hypothesis, coupled with the previous observations, lead to my research goals:

- Associate the topic of Wikipedia articles with concepts in the Cyc ontology
- Use the ontology itself to organize most of the work

1.2 Goals of the Research

The overall goal of the research is to create a system that given a set of specific examples and their initial mapping, can automatically construct recognizers for more abstract classes, and using those recognizers to classifier new inputs with respect to the hierarchy.

An additional benefit will be the ability for the system to do just-in-time-learning. The system starts by hypothesizing new articles are members of the most general classes and depending on their classification the system places new hypothesis on an agenda for further processing. The system may hypothesize that an article is a member of a class that it does not have a classifier for. At this point it will compile a class recognizer from the existing examples, classify the article, store the newly created recognizer for future use, and if the article was a member of the hypothesized class then that classes' children will be added as new hypothesizes to be explored.

The benefit of the overall framework is:

- The system only generates recognizers that are necessary
- The system only generates them in-time
- New classifiers could be generated at anytime
- Items needing to be classified need not be limited to text, but could include images, audio or any other classifiable data source

The output of the system is a list of positive and negative classifications that apply to the input.

Note also that the system is not limited to just the text classifier, but also can use other classifier methods such as Weka, neural networks, genetic algorithms/genetic programming, etc., to build classifiers. The only requirement is the ability to associate a set of known bindings, and a somewhat hierarchal ontology of classes. Note also that the system could defer the classification of a new object until a required number of examples supporting are available. This way the system can "know what it does not know," and schedule classifier creation for a time when it knows more.

In order to meet these goals I performed the following:

- Implemented the required infrastructure, consisting of the Wikipedia, Cyc, and the Perl based classifier
- Developed access methods between Perl, Wikipedia, and Cyc
- Developed the set of relations to represent Wikipedia articles in Cyc
- Developed a text classifier that can classify wither or not an article is relevant to a particular Cyc concept

- Developed a frame work to search for the proper classification for articles in a top down manner

1.3 Required Resources

To complete the project I used the following resources:

Hardware:

- Current Pandor and associated computers at Daxtron Labs

Software:

- Perl – readily available from the internet
- ResearchCyc – used in previous projects and available from Cycorp via http://researchcyc.cyc.com
- Perl::LWP – a CPAN module that allows Perl to interact with Web servers
- Perl::Tie::Persistent – a CPAN module that saves Perl hashes to disk
- RAR – a file compression utility to compress the text classifier files
- MediaWiki – the software that runs the Wikipedia
- WAMP – Windows Apache MySQL PHP module (used to support MediaWiki on Win32 machines)
- Wikipedia Special:Export – the XML interface used to access the text of an article

Data requirements:

- The Wikipedia Database – the XML download of the Wikipedia

1.4 Milestones

- Initial installation of infrastructure
- Development of interfaces between Perl , Cyc and Wikipedia

- Insertion into Cyc of initial seed articles
- Development of text classifier
- Develop autogenerator of text classifier using Cyc (i.e., given a Cyc class generate a text recognizer for that class)
- Develop a wrapper process to walk an article through its classification
- Complete evaluation of the system

1.5 Evaluation Criteria

The goal of the research is to determine if the use of an ontology to organize training instances to add information that will ultimately go into the ontology is a valid concept worth further exploration. While the overall procedure may be sound, its operation will depend on the performance limits of the recognizers it can develop. To determine this, statistics on the accuracy of the text classifiers as they are trained was collected and analyzed.

1.6 Research Contribution

My main purpose was to create a system that given a hierarchy can automatically generate recognizers to place new instances in that hierarchy. In particular, this hierarchy is defined by the Cyc ontology plus inference developed to explore it, and the new instances are the topics of Wikipedia articles.

This work developed and tested an overall framework that combines semantic ontologies like Cyc, WordNet and the Semantic Web with text classification tools to test the core of an automated just-in-time classifier compiler technology. This should be unique and be a contribution to the field. Existing approaches utilize total creation of

the set of recognizers, or attempt detailed extraction. In this method the text of a whole descriptive document is used to make assertions about the thing it describes. The overall framework is general enough to have wider applicability.

1.7 Thesis Outline

The rest of the thesis is organized as follows:

Chapter 2 provides necessary background material, including describing the overall problem of mapping instances to an ontology, the Cyc ontology and inference engine, and the Wikipedia.

Chapter 3 describes the construction language models for text classification, and their application towards this task.

Chapter 4 details the process that oversees model construction and article classification.

Chapter 5 outlines the performance of the system.

Chapter 6 relates this work to that of others.

Chapter 7 gives my conclusions and highlights possible future work.

Examples of system performance can be found in Appendices, along with key system definitions.

CHAPTER 2

KNOWLEDGE SOURCES: CYC AND WIKIPEDIA

2.1 Overview

The primary information systems that are the focus of this project were Cyc and Wikipedia. Cyc is a large, controlled semantic ontology expressed in a formal higher order predicate logic. Wikipedia is a large, ad-hoc, volunteer-generated natural language encyclopedia. Cyc was created to provide the baseline knowledge necessary to understand an encyclopedia, and Wikipedia is a freely available electronic encyclopedia. In this chapter I briefly describe both.

2.2 Cyc

2.2.1 Introduction to Cyc

Due to the success of expert systems like DENDRAL, MYCIN and XCON in the early 1980s, there was a surge in enthusiasm for artificial intelligence research [Lindsay et al., 1980; Buchanan et al., 1984; Sviokla, 1990]. It has been argued that despite the fact that there was a defined path to the construction of expert systems, a lull occurred in their use due to their inflexibility [Friedland et al., 2004] and sharp loss of their expert ability when asked to reason about situations that varied slightly from their original design specifications. The Cyc project [Lenat et al., 1983; Lenat, 1995] was initiated to overcome the brittleness issue by providing computers with a store of formally represented general commonsense knowledge in which domain specific expert knowledge could also be embedded and to which programs can draw on when faced with situations partially or wholly outside their original domain. Human "Cyclists"

have been manually entering knowledge into the Cyc knowledge base in various ways over the last twenty years. Having expended approximately 900 person-years of effort they currently have represented over two million facts and rules about more than 300K entities and types of entities, 26K relationships, and close to three million assertions and rules. According to Cycorp, Cyc is "the world's largest and most complete general knowledge base and commonsense reasoning engine."

2.2.2 The Cyc Knowledge Base and Inference Engine

The inference engine supports deductive, abductive and inductive inference over the knowledge base by integrating more than 700 specialized reasoners for commonly occurring classes of sub-problems. The knowledge base (KB) is intended to support unforeseen (and even unforeseeable) future knowledge representation and reasoning tasks by providing facilities to represent and reason over first and n^{th}-order predicate logic [Ramachandran, 2005]. It also supports the ability to segment the knowledge into local inheritable contexts [Guha, 1991] called *microtheories*. Microtheories can be mutually consistent with their parent contexts but can contradict their siblings. This allows multiple, possibly contradictory, viewpoints to be represented simultaneously.

2.2.3 Natural Language Processing

In addition to the logical ontology/KB, Cyc contains natural language (NL) processing tools and information. A well-developed English lexicon, containing the knowledge about syntax and semantics, allows it to translate between its formal representations and English. Cyc can also link to WordNet [Miller et al., 1990].

2.2.4 Ontology Content

The ontology provides a wide range of categories in order to be relevant across many domains. A fundamental distinction is made between collections and individuals. This is relevant to because different domains have different units of focus. Specific individual people, place and events tend to be the focus of history, while science tends to express information as properties ascribed to entire classes or conditions. The logical predicates provided for a domain indicate which level knowledge is expressed at.

An example of the type of data included in the Cyc KB is shown below for the concept of "Lobster":

Figure 1: Cyc lobster example.

2.2.5 The Upper, Middle, and Lower Cyc Ontology

For those who work with Cyc, the KB is traditionally subdivided into an upper, middle, and lower ontology. At each level different generality of the information is provided. The upper ontology contains broad, abstract, or highly structural concepts, and provides the highly referenced definitional core of the system. It is designed to represent concepts such as temporality, mathematics, and relationship types.

Abstractions that have high reusability, yet are not nearly universally applicable, are stored in the middle ontology. Examples include geospatial relationships, broad knowledge of human interactions, or everyday items and events. While providing "commonsense" they do not represent knowledge needed by all applications. It contains the type of knowledge "everyone" has a high probability of knowing, yet would use only a faction of it at any time.

Domain-specific knowledge is contained in the lower ontology, and contains information about a specific field or individuals. This section of the ontology has the largest "mass," but has the lowest reusability. It typically aims to contain knowledge everyone familiar with the particular field knows, but in general not "everyone" can be expected to be familiar with that field.

```
(comment Event "An important specialization of Situation, and thus also of IntangibleIndividual and
TemporallyExistingThing. Each instance of Event is a dynamic situation in which the state of the world
changes; each instance is something one would say 'happens'. Events are intangible because they are
changes per se, not tangible objects that effect and undergo changes. Notable specializations of Event
include Event-Localized, PhysicalEvent, Action, and GeneralizedTransfer. Events should not be confused
with TimeIntervals.")

(isa Event TemporalStuffType)
(isa Event Collection)
(quotedIsa Event PublicConstant-CommentOK)
(quotedIsa Event VocabularyConstrainingAbstraction)
(genls Event Situation)
(disjointWith Event PositiveDimensionalThing)
(genls InstantaneousEvent Event)
(genls HelicopterLanding Event)
inferred knowledge
(genls (BecomingFn Intoxicated) Event)
(relationExistsAll victim Event Victim-UnfortunatePerson)
```

Figure 2: Upper ontology example—*Event*.

For every instance of the collection *Victim-UnfortunatePerson,* there exists an *Event* in which that person was the victim—i.e., an event for which these statements hold:

```
(victim ?SOMEVICTIM ?SOMEEVENT)
(isa ?SOMEVICTIM Victim-UnfortunatePerson)
(isa ?SOMEVICTIM Event))
```

```
(comment Situation "A specialization of both IntangibleIndividual and TemporalThing. Each instance of
Situation is a state or event consisting of one or more objects having certain properties or bearing certain
relations to each other. Notable specializations of
Situation are Event and StaticSituation; it is disjoint with SomethingExisting.")
(genls Situation TemporallyExtendedThing)
(genls Situation TemporallyExistingThing)
(genls Situation IntangibleIndividual)
(disjointWith Situation SpatialThing)
(implies
(and
(isa ?AGT Agent-Generic)
(causes-Underspecified ?AGT ?EVT)
(isa ?AGT Situation))

(causalActors ?EVT ?AGT))
```

Figure 3: Upper ontology example—*Situation*.

If there is a situation that is caused (in some sense) by some agent, that agent plays the role of causal actor:

```
(implies
(and
(isa ?SIT Situation)
(providesMotiveFor ?SIT ?AGENT ?EVENT-TYPE ?ROLE))
(increases-Generic ?SIT
    (relationExistsInstance ?ROLE ?EVENT-TYPE ?AGENT) likelihood))
```

If some situation provides a motive for an agent to play a certain role in some kind of event, the likelihood of that event occurring increases.

```
(comment SocialGathering "A specialization of SocialOccurrence. Each instance of SocialGathering is an
intentional social gathering of people who have the same or similar purposes in attending, and in which
there is communication between the participants.
Specializations include BabyShower, Carnival, and Rally. Note that a group of people waiting to board an
elevator is not typically a SocialGathering, even though they share a common purpose, since they are not
expected to talk to each other.")
(disjointWith SocialGathering SingleDoerAction)
(disjointWith SocialGathering ConflictEvent)
(disjointWith SocialGathering IntrinsicStateChangeEvent)
(keStrongSuggestionPreds SocialGathering dateOfEvent)
```

Figure 4: Middle ontology example—*SocialGathering*.

Although it is not semantically required, it is likely that getting a *dateOfEvent* assertion for any given instance of *SocialGathering* would be appropriate or desirable.

```
(requiredActorSlots SocialGathering attendees)
```

Translation: In every social occasion something must play the role of attendees.

```
(comment ChemicalReaction "A collection of events; a subcollection of PhysicalTransformationEvent. Each
instance of ChemicalReaction is an event in which two or more substances undergo a chemical change,
i.e., some portions of the substances involved are transformed into different ChemicalSubstanceTypes.
The transformations are brought about by purely chemical (including biochemical) means which affect
chemical bonds between atoms in the molecules of stuff. Examples of ChemicalReaction: instances of
CombustionProcess; instances of Photosynthesis-Generic.")
(keGenlsStrongSuggestionPreds-RelationAllExists ChemicalReaction catalyst)
(genls ChemicalReaction PhysicalTransformationEvent)
(genls CombustionReaction ChemicalReaction)
(genls RNASplicingProcess ChemicalReaction)
(genls ExothermicReaction ChemicalReaction)
(genls ChemicalSynthesis ChemicalReaction)
```

Figure 5: Lower ontology example—*ChemicalReaction*.

2.2.6 The CycL Language

A number of extensions to first order predicate logic are included in CycL [Ramachandran et al., 2005]. It provides for quantification over predicates, function and

sentences, and predicates can take other predicates as arguments. It provides quoting in a form that allows it to differentiate between knowledge involving a concept, and knowledge about the *term* for that concept. This gives the system the ability to both represent that "all dogs are mammals," and that the term used for dogs (in Cyc "*#$Dog*") was introduced by a certain person at a certain time.

Quoting and higher-order extensions allow the representation of the semantics of the language's terms to be represented within the same language as all other knowledge. This all means that all knowledge has a common representational basis, is mutually accessible to inference, and increases expressivity. It also allows easier implementation of "rule macro predicates," which allow for compact expressions of relationships that would otherwise require rules.

2.2.7 The Semantic Constraint Vocabulary

At the core of the Cyc ontology is the taxonomic knowledge that defines the class membership of terms via the *isa* relationship, and the subsumption relationships between those classes via the *genls* relationship. The primary knowledge about the intended meaning of predicates and interaction of their arguments are defined by these taxiomatic definitions. Predicates also participate in subsumption relationships via the *genlsPreds* relations, such that if a relation holds for set of predicate arguments then it also holds for a different predicate. An example of such a relation would possibly be between *friends* and *knowAbout*. If *friends(X,Y)* holds then *knowAbout(X,Y)* also holds.

A large body of knowledge also encodes the semantic correctness an applicability of the predicates, and thus defines the grammar of legal CycL statements. Thus the first argument of *friends* can be limited to the domain of *IntelligentAgents*. If a predicate was used without the first argument being an *IntelligentAgent* then the statement would not be semantically ill formed. In this way syntax mirrors semantics. According to the definition of the grammar of predicates, the constraints of "common sense" are enforced on the use of the relationships and predicates. It quickly flags semantically impossible statements. (That is, until the semantics are redefined).

One example used is to logically encode the English sentence "Julia gave birth to a prime number." It is syntactically well-formed English sentence but it expresses a concept that the predicate *biologicalMother* should not be allowed to be true about. So the predicate *biologicalMother* is defined as a binary predicate that relates an animal to the female animal that gave birth to it:

```
(isa biologicalMother IrreflexiveBinaryPredicate)
(arg1Isa biologicalMother Animal)
(arg2Isa biologicalMother FemaleAnimal)
```

The first formula expresses that nothing can bear the relation *biologicalMother* to itself. The predicate *biologicalMother* belongs to the set of irreflexive relations. While the expression could be represented axiomatically, this format allows the expression of knowledge shared with other *irreflexiveBinaryPredicates*.

The two other formulas state that the predicate *biologicalMother* can accept an animal as its first value and a female animal as its second value.

The predicates *genls* and *disjointWith* also provide taxonomic knowledge of a different form. *Genls* expresses inheritance among types, classes and structures the ontology, while *disjointWith* states that two collections do not share any members. The predicate *isa* is another primitive relation, which expresses individual membership in classes (CycL collections).

2.2.8 OpenCyc

OpenCyc is a freely available subset of the knowledge base. It also includes a free-to-use executable Knowledge Server that includes an inference engine and other tools for browsing, manipulating and using the content of the knowledge base. Information and downloads are at http://www.opencyc.org.

In its current version, OpenCyc includes a definitional ontology that mirrors those of full Cyc, and a subset of the assertions OpenCyc provides access to a foundation of concepts that can be used and easily extended. OpenCyc also provides CycL, the logical language used to support the Cyc ontology. The definitional vocabulary provided in OpenCyc summarizes the taxonomic information that was discovered to be required over the course of building the Cyc KB. Like most knowledge, it is a form of compiled experience.

2.3 Wikipedia as a Source of Knowledge

Wikipedia is a large, community-created encyclopedia. At the time of writing it contained 6 gigabytes of text, 80 gigabytes of images and 1.3 million articles. In June

2006, it contained 511 million words, had an average of 3113 bytes per article and had an expansion rate of 2106 articles per day.

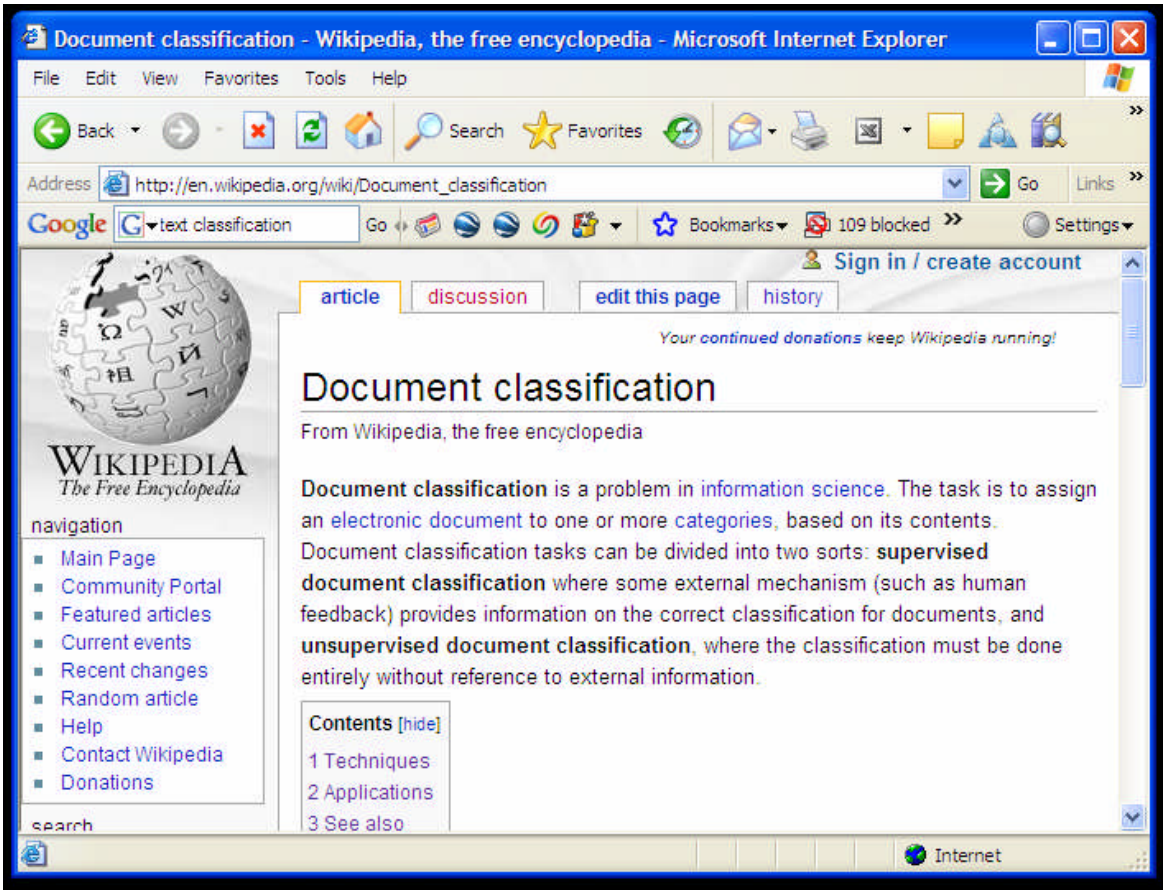


Figure 6: Wikipedia example web page.

The Wikipedia was all written using a Wiki. A Wiki is software tool that allows collaborative editing of hypertext documents. Using a Wiki people can create and edit hypertext documents using a relatively simple syntax. This simple syntax lowers the learning curve required to generate articles, and thus allows a larger author base. At the time of writing there were 13,000 registered contributors to the Wikipedia system.

The set of Wikipedia sites is hosted using the open source project MediaWiki. MediaWiki is Wiki software written in PHP and uses a MySQL database. It is used to host many encyclopedia and dictionary sites including Wikipedia.

Given the breath of coverage of the Wikipedia it is natural to wish to utilize it as a knowledge source. One of the stated guidelines of Wikipedia authors is that the articles be descriptive and written as neutrally as possible. While not always the case, it is does form a pool of documents more uniformly focused than that of the open web. That is, the "signal to noise ratio" is higher when seeking information describing a specific item.

Other projects have sought to access the information in Wikipedia. One of the more notable ones is the Semantic Wikipedia project (http://www.semanticwiki.jp). The goal is to provide the ability to manually annotate Wikipedia articles with semantic information that could be automatically extracted. In this way, while editing the Wikipedia, information needed to transform it into an ontology is captured. However, this process is not automatic, depending on the human editing of every article.

2.4 Summary: Two Great Things that Should Go Great Together

As two large-scale reference works, Cyc and Wikipedia would seem to be naturally complementary. Over the years Cyc was developed to provide a formal basis for understanding the contents of an encyclopedia, and the Wikipedia is a large encyclopedia in computer readable format. As such the two should go together quite nicely.

However, the first step of integrating the two is to match the articles in the Wikipedia to the concepts in the Cyc ontology. Doing this automatically was the focus of the research.

CHAPTER 3

LANGUAGE MODELS FOR TEXT CLASSIFICATION

3.1 Introduction

Assigning a document to a category or class is the primary function of text classification. Many methods exist for text classification and language models are well suited for some versions of this task. This lead to the following questions:

1. What are language models?
2. How do they work?
3. How can they be used for our task?

A language model approach would view determination of text class as a sub-language identification problem. Language models are commonly used to discriminate one language from another. This feature can be used to identify the characteristic language used to talk about a given domain. For instance, sports, medicine, computers, aviation and the various sciences each have their own way of communicating about the elements in their field. In this project we use this ability at a more fine grain level to determine if a given article was generated to describe something that would fit within a given section of the target ontology.

The probabilities returned by a language model depend on the language being modeled, and the features one collects statistics on. Here the model will be of "the English used for describing things of type C," and the features we examine will be n-gram bases: sequences of 1 to n words taken from a sliding window over the text.

3.2 Computing Probabilities of Class Membership with Naïve Bayes-style Classifiers

Given document D, and a set of classes $C_1,\ldots C_n$, a Naïve Bayes classifier computes the posterior probability that a document D belongs to each class C_i , $P(C_i|D)$, and classifies the document as the class with the highest probability value. Using Bayes' rule one can compute the posterior probability as:

$$P(C_i \mid D) = \frac{P(D \mid C_i)}{P(D)} \times P(C_i) \tag{3.1}$$

However, P(D) is constant can be ignored for ranking purposes:

$$P(C_i \mid D) \approx P(D \mid C_i) \times P(C) \tag{3.2}$$

$$P(C_i) = \frac{N_i}{N} \tag{3.3}$$

where N = total number of training documents; and N_i = number of training documents in class C_i.

$P(D|C_i)$ = the probability that document D is in class C_i

D is made up of words $d_1,\ldots d_m$

$$P(D \mid C_i) = \prod_{j=1}^{m} P(D_j \mid C_i) \tag{3.4}$$

$$P(C_i \mid D) \approx P(C_i) \times \prod_{j=1}^{m} P(D_j \mid C_i) \tag{3.5}$$

Converting the equation to do computations in log space (for efficiency and to prevent overflow/underflow) generates:

$$\log(P(C_i \mid D)) \approx \log(P(C_i)) + \sum_{j=1}^{m} \log(P(D_j \mid C_j)) \tag{3.6}$$

The key value that needs to be computed is $P(D_j \mid C_i)$ or the probability of a word D_j in a document occurring if it the document was of class C_i:

$$P(D_j \mid C_i) = \frac{1 + Count(D_j, C_i)}{|V| + N_i} \tag{3.7}$$

where $|V|$ = number of terms in the vocabulary; Count (D_j, C_i) = number of times that word D_j occurs within the training documents of C_i; and N_i is the total number of words in that appear in class C_i.

A document D is assigned to the class C' with the highest probability of generation:

$$C' = \arg_{C_i} \max\left[\log(P(C_i \mid D))\right] \tag{3.8}$$

One of the benefits of using a language models is the use of smoothing techniques, to address the zero occurrence problem. The purpose of smoothing is to provide non-zero probabilities to events, such as the occurrence of an n-gram, and thereby improve the maximum likelihood estimation. Since the probability calculation is a chain multiplication, a zero would result in a zero estimation, which would be false if the zero was returned for both the positive and negative classes.

For the classifiers implemented I used a smoothing estimate for words unseen in the training:

$$P(Word_{unseen} \mid C_i) = \frac{1}{\sum Positive\_class\_words + \sum Negative\_class\_words} \quad (3.9)$$

$$P(Word_{unseen} \mid C_i) = \frac{1}{\sum_{j=1}^{|V|} Count(Word_j, C_{positive}) + \sum_{j=1}^{|V|} Count(Word_j, C_{negative})} \quad (3.10)$$

That is, given no evidence from the training set, the probability is divided evenly between the two classes. This smoothing formula was found useful in n-gram language discrimination.

3.3 Measuring Classifier Performance

For any classifier there are a set of standard measures to evaluate the performance of a classifier. I use the standard recall, precision and f-measure to evaluate the performance of each classifier as it is trained. Given a Cyc constant Ci, the classifier must determine if a document D belongs to either class +Ci or -Ci, where +Ci means it the article should be associated with the constant Ci, and –Ci means it should not.

Given:

TP = true positives, the number of documents correctly identified as +Ci

FP = false positives, the number of documents incorrectly identified as +Ci

TN = true negatives, the number of documents correctly identified as -Ci

FN = false negatives, the number of documents incorrectly identified as -Ci

Precision, recall and f-measure for +Ci and –Ci are defined by:

$$Precision(+C_i) = \frac{TP}{TP + FP} \tag{3.11}$$

$$Recall(+C_i) = \frac{TP}{TP + FN} \tag{3.12}$$

$$F - measure(+C_i) = \frac{2 \times Precision(+C_i) \times Recall(+C_i)}{Precision(+C_i) + Recall(+C_i)} \tag{3.13}$$

$$Precision(-Ci) = \frac{TN}{TN + FN} \tag{3.14}$$

$$Recall(-C_i) = \frac{TN}{TN + FP} \tag{3.15}$$

$$F - measure(-C_i) = \frac{2 \times Precision(-C_i) \times Recall(-C_i)}{Precision(-C_i) + Recall(-C_i)} \tag{3.16}$$

Precision provides the percentage identified as the target class that are actually of the target class. Recall provides the percentage of the actual target class that was correctly identified. In most applications there is a tradeoff between precision and recall. F-measure [Lewis et al., 1994] is used to combine the precision and recall into a single value.

```
=== Detailed Accuracy By Class ===
  TP Rate   FP Rate      Precision         Recall    F-Measure         Class
   1.0000     0.0000       1.0000         1.0000      0.9286       + #$NaturalThing
   0.8889     0.1111       0.8889         0.8889      0.9412       - #$NaturalThing
=== Confusion Matrix: #$NaturalThing 1  ===
   +Class       -Class       <-----Classified as
TP: 0013      FP: 0000     + #$NaturalThing
FN: 0002      TN: 0016     - #$NaturalThing
==========TEST ERROR============
Correctly Classified Instances   29          93.5483870967742 %
Incorrectly Classified Instances  2          6.45161290322581 %
Total Number of Instances        31
```

Figure 7: Example of evaluating the performance of a classifier on the *NaturalThing* concept.

By having a performance profile for each classifier, it is possible for multiple classification methods to be used on a per-class basis.

3.4 Training the Language Models

Language models are generated to cover text documents that are positive and negative instances of the target class. This creates an initial training set which is iteratively added until either a performance criteria is met (accuracy > 90%) or a maximum number of trials (n=8) is reached.

```
Constant: positivePCWExample.
isa: BinaryPredicate.
arg1Isa: Thing.
arg2Isa: Thing.

Constant: negativePCWExample .
isa: BinaryPredicate.
arg1Isa: Thing.
arg2Isa: Thing.

f:(implies
(and      (topicOfPCW ?W ?C)
          ( or
 (isa ?C ?TargetConcept)
                    (genls ?C ?TargetConcept)  ) )
(positivePCWExample ?TargetConcept ?W) ).

f:(implies
(and      (topicOfPCW ?W ?C)
          (not ( or
(isa ?C ?TargetConcept)
                    (genls ?C ?TargetConcept) ) ) )
          (negativePCWExample ?TargetConcept ?W) ).
```

Figure 8: KE-Text definition of *positivePCWExample* and *negativePCWExample*.

Figure 8 above shows the definition for *positivePCWExample* and *negativePCWExample*. They are written in KE-Text, a format used to enter CycL assertions from text files. PCW is shorthand for propositional conceptual work, an abstract work containing in some part propositional information. This includes things that contain words that express propositions like literary works. The syntax of KE-Text is fairly simple to allow CycL to be entered quickly. Each line contains either an operation or a predicate, followed by a ":" separator character, followed by any arguments. Comments begin with double semi-colons. The operator "f:" indicates that a CycL formula follows and will assert the CycL sentence that comes after the ":" to the KB; it is often used to specify rules or more complex formulae:

```
Constant: positivePCWExample.      ;;Create a constant term positivePCWExample
isa: BinaryPredicate.              ;; The term positivePCWExample is a BinaryPredicate
                                   ;; (assert  (isa PositivePCWExample BinaryPredicate) )
arg1Isa: Thing.                    ;; Its first argument is a Thing
                                   ;; (assert (arg1isa PositivePCWExample Thing)  )
```

Two pools of examples are used initially. The first is derived from a reference set of specific instances. Each reference instance is tested for class membership and assigned to the proper set. Thus members of this reference set are in all classifiers. Cyc is then queried to obtain known positive and negative instances for the target classes that have an article using the definition given above.

Given this set of examples the system performs either a complete sample or random sample depending on if enough samples exist, until sufficient text is collected to form a minimal model. This is taken as 350K for the positive class and 700K for the negative class. N-gram statistics (unigram, 2-,3-,4-grams) are collected for the text of each class.

Once the initial model is created it is tested against a random sample from both classes, and the performance is measured. During this testing phase those articles that are misclassified are noted for later inclusion into the language model. If performance is below acceptable limits and additional trials remain, the misclassified articles are added to the language model and retested. However, if enough trials have passed or the model is sufficiently accurate, then the system will store the classifier and use it to process the query document.

If the topic of a new document is classified as belonging to the Cyc class C_i, then the classifier was for that Cyc concept, and Cyc is queried to find acceptable children of

class C_i. The children not already on the queue are added to the queue and a new round of classification or classifier construction can begin.

3.5 Examples and Walkthrough

Appendix B contains traces for processing an article on Ronald Reagan. The first section shows the statistics of various classifiers when trained on Cyc concepts visited while trying to classify the article. These include: BiologicalLivingObject (T), Place (F), Product (F), TimeInterval (T), NaturalThing (T). The second section details the training collection and training process of "CombustionInstrument." In particular it shows the selection process for the positive and negative examples.

3.6 Analogy to Other Text Classification Methods

An analogy can be made between the task of classifying ontology classes and spam-classification. The problem in spam-classification is to detect a very fuzzy abstract concept (useful to the recipient versus spam) using a minimum of analysis, but looking at the content of the document as a whole. Bayesian analysis methods are often used for the spam classification task. Several methods from the successful spam classifier CRM-114 were tested and included in the system. CRM-114 is a toolkit used for filtering e-mail spam and classifying data using statistical methods. CRM-114 has achieved higher performance ratings using a form of simplified Hidden Markov Model embedded in an n-gram framework [Yerazunis et al., 2005].

3.7 Feature Generation

The articles were selected from a local copy of the Wikipedia, and accessed via the *Special:Export* function. *Special:Export* returns an XML document from the MySQL database of the Wikipedia article, and everything but the article text is stripped. XML/HTML tags and punctuation are removed along with stop words. No language-specific preprocessing or word stemming is performed. The text is broken into word tokens via Perl's split function and n-grams are generated.

First, sparse binary polynomial hashing (SBPH) was tested. SBPH is a method to generalize recognition of n-grams used by CRM-114. SBPH uses a sliding window of size n over the set of tokens. For each window, all possible in-order combination of the tokens are generated. So, for a given window of size n the system generates 2^n-1 features. For the sequence "Buy Viagra Now" SBPH generates unigrams {"Buy," "Viagra," "Buy Viagra," "Now"}, bigrams {"Buy Now," "Viagra Now"} and trigram {"Buy Viagra Now"}. Generation of these additional features is designed to cover plausible variations used to communicate a concept.

Experimentation showed that basic SBPH did not improve recognition on this task.

Another method used by CRM-114 is called exponentially superinceasing model weighting (ESMW). ESMW is based on the idea that longer n-grams should be more discriminative than shorter ones. Thus, matching an n-gram should be exponentially more relevant than matching an (n-1)-gram; the sum of (n-1)-grams should not outweigh an n-gram.

N-gram	ESM-Weight
1	1
2	4
3	16
4	64
5	256
6	1024

Table 1: ESM weights.

The use of ESMW was found to improve performance of the text classifiers. However, the use of ESMW changes the probability estimate into a scoring function. The use of ESMW does allow the effects of Markovian chaining to occur. Longer chains of words in a corpus can overrule single words or shorter chains. This gives a system the ability to perform non-linear filtering.

3.8 What are We Classifying?

The language models constructed can return the probability that the generator of a sequence of words observed in some document D was describing something from category C_i. This is due to the fact that the classifier was trained on examples of documents that describe things in category C_i. Just as Cyc makes the distinction between the class in the real world and the symbol in the system used to describe it, we make the distinction between the item in the real world and the Wikipedia article used to describe it.

As a starting point the system assumes the following for each article:

```
(ThereExists ?Something
  (and
    (isa ?Something Thing)
    (equal ?Article <some-wikipedia-article-id>)
```

```
(isa ?Article WikipediaArticle)
(topicOfConceptualWork ?Article ?Something)
(url ?Article <url>)…)
```

Translation: *There exists something that a known Wikipedia article is about.*

Each classifier then can add information of the form:

```
(and
    (isa ?Something <some-Cyc-Concept>)
    ….)
```

Translation: *And that something described by the text of the article also belongs to the collection of <some-cyc-concept>.*

With simple extensions we could attempt to train classifiers for other types of Cyc formula. For instance, one could train classifiers to make assertions of the form:

```
(conceptuallyRelated ?Something <some-Cyc-Concept>)
```

Cyc currently has approximately 4000 binary predicates that fit this format, and additional relationships can be gerated as required.

So, given an article on Ronald Reagan a CycL assertion of the following form could be built up:

```
(ThereExists ?Something
    (and
        (isa ?Something Thing)
        (equal ?Article Wikipedia wikip-RonaldReagan )
        (topicOfConceptualWork ?Article ?Something)
        (url ?Article http://localhost/cyclopedia/RonaldReagan )
        (isa ?Something Leader)
        (isa ?Something MaleHuman)
        (isa ?Something Politician)
        (isa ?Something UnitedStatesPresident)
        ….))
```

Figure 9: CycL equivalent of processing the Ronald Reagan article.

As each classifier processes the text and makes its determination it adds its own fragment to the description of the thing described.

While similar to text classification for building a Web directory like Yahoo or DMOZ [Mladenic et al., 2004], this task is different. In the web directory case the focus is on the pages and their contents, not the object they describe (if any). Often the pages are offering services or products and may or may not be descriptive of some external to the page object. Often the pages that are being classified are hubs of links *to* information on the topic and not direct sources of information *on* a topic. In this task I was interested in determining information about what a page describes and not information about the page itself.

3.9 Summary

After exploring the various options for language models, a reasonable model was determined and used as the core classifier in the system. Any additional work in improving text classifier performance will have a direct benefit in the overall system.

See the traces for examples of classifier performance improving for easy-to-learn classes, and stagnating for hard-to-learn ones. Utilizing multiple-classifier types on a per-class basis may be of benefit. Cyc's ontology would allow such information to be associated on a per-class basis allowing the system to apply inference to the task of deciding which of a set of classifiers the best to use is.

CHAPTER 4

SELF-SUPERVISED LEARNING OF TEXT CLASSIFIERS VIA SEARCH

4.1 Overview

The primary task of the system is to identify which classes in a reference ontology can be said to be about the thing described in a target document. For example, by analyzing the Ronald Reagan article the person described in it can be recognized to belong to the Cyc categories *Leader*, *MaleHuman*, *Politician* and *UnitedStatesPresident*. The reference ontology graph can provide a map guiding classifier generation, organizing example articles and ensuring that the classifiers created match in someway the model of the ontology creators (e.g. humans) versus a totally unsupervised process.

4.2 The Search Space

The reference ontology used is the Cycorp Cyc ontology. Cyc's ontology covers a large cross-section of how humans segment their universe into objects types and concepts.

In all there are 30,000 collections or classes in Cyc. A subset of 418 collections were selected and marked as belonging to the *TextClassifierClass*. First for every class the number of subsumed articles was calculated and the list sorted to find the classes with the most articles. From that list the most frequent corresponding to the article writer's boundaries were identified. Cyc has a fine grain structure that does not always align with article writer's classifications. For instance while there are several distinctions for sub-aspects of geography most of the articles focus on human identifiable "Places." In the initial system, passing through several geography nodes only to reach *Place* was

seen as unnecessary. Such a heuristic would be re-examined in future systems. To this were added 156 classes provided by Larry Lefkowitz at Cycorp based on work done to determine a maximal non-overlapping cover of non-metaphysical concepts to aid in knowledge collection from humans. The list was generated to provide a list of O(100) that would allow a person to quickly describe a set of key relevant features or classes when introducing a new object to Cyc. Additional rationale for selecting the 418 initial categories is given in Section 5.2.2, Restrictions on trainable classifiers.

4.3 The Search and Construction Process

The basic process is a breadth-first search over the search space starting at a set of initial nodes, moving outwards towards the articles:

- Given an article, the system places an initial set of class names on the queue to test. In this case we used some of the higher-level Cyc categories that had a large number of articles as instances in the upper ontology. This initial queue consisted of: *#$BiologicalLivingObject*, *#$Individual*, *#$Collection*, *#$Place*, *#$Agent-Generic*, *#$Artifact-Generic* and *#$Product*.
- *Dequeue Process*: If there are elements on the queue the system removes the first one for processing. Otherwise, terminate and report the classes the article was found to be a member of.
- Determine if the class has a classifier associated with it. If so, then recall the classifier. If not, then construct one.
    - Construct a positive and negative set of articles by asking Cyc to classify members of a reference set, and random sample of the initial set. (See Figure 2 for an example).
    - Using the positive and negative set, construct an n-gram based language model.

    - Iteratively test the performance of the language model, adding misclassified articles until either a target performance level is reached, or a maximum number of iterations have been performed.

- Save the classifier.
- Classify the article using the retrieved or generated classifier.
- If the article is not classified as a member of the class then go to the *Dequeue Process.*
- If the article is a member of the tested class then ask Cyc for the nearest children of the tested class that can have classifiers constructed for them. Add those not already on the queue to the queue (the ontology is a DAG and may have multiple paths to the same node) and go to the *Dequeue Process.*

The search process is more tolerant of false positives than false negatives, yet has the ability to recover from some misclassifications. First, a false positive will result in incorrect expansion of a node. This will result in multiple children of that node being placed on the queue and each must be verified to progress further. Since the children are more specific their acceptance criteria will be stricter. Thus filter strength increases with depth in the ontology and incorrect paths quickly decay. In the case of false negatives a correct node is not expanded and its children are not included. Since the Cyc ontology is a DAG with multiple paths to a given leaf, a given leaf can be reached via multiple means. As long as some of those paths remain open, the possibility of final classification exists. Once an article is assigned to a class (either leaf or interior) then all the information on that class can be inherited, including the misclassified elements. This should provide the classification system with additional levels of robustness, allowing it to "fill in the blanks" which is one of the original goals of Cyc. Also, the incorrect classifier can be identified and trained on the new instance.

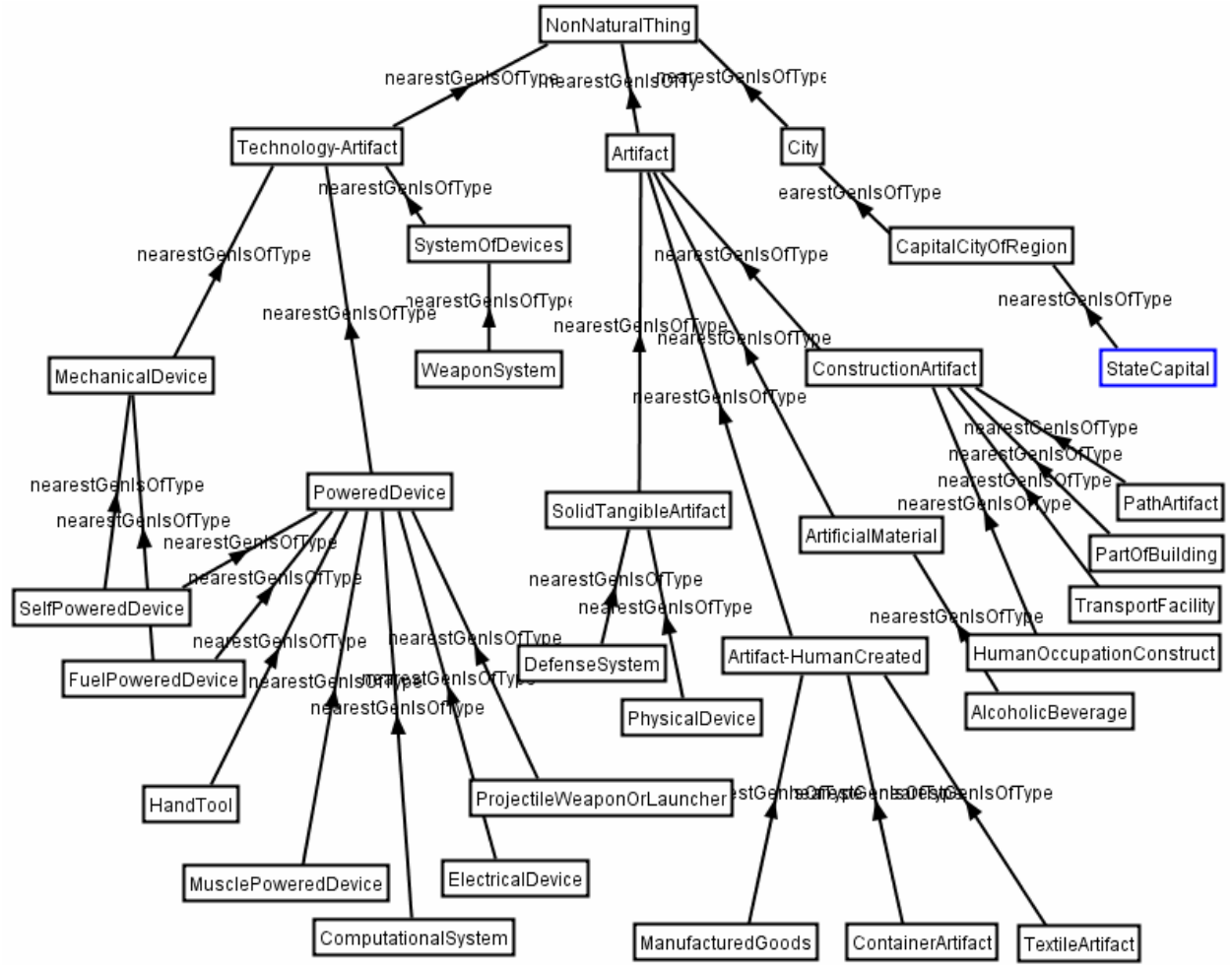


Figure 10: Partial graph under *NonNaturalThing*.

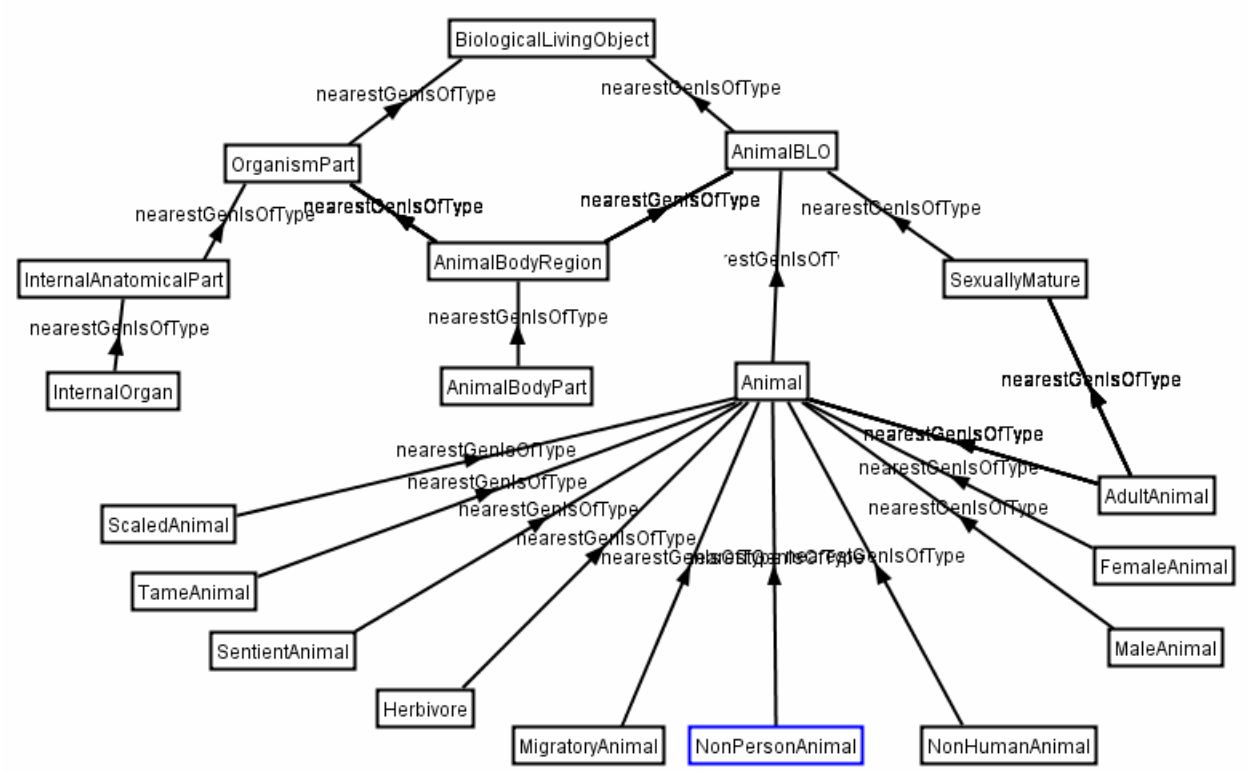


Figure 11: Partial graph under *BiologicalLivingObject*.

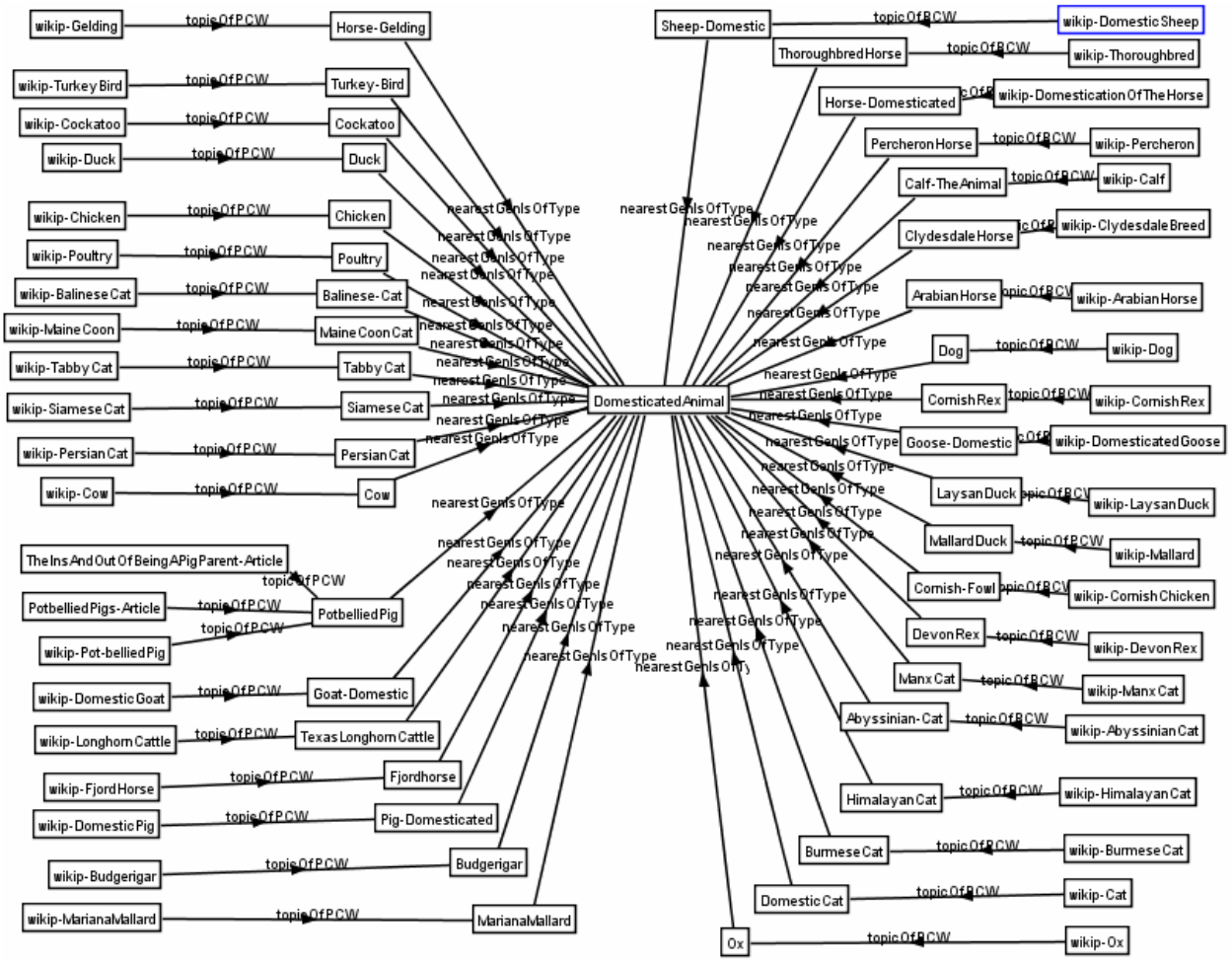


Figure 12: Subset of positive articles for *DomesticatedAnimal*.

4.4 Association of Articles to the Ontology

Each initially mapped article is represented in the ontology, and stored in the *WikipDataMt* microtheory. The constant used to represent the article is derived from the article name, and is declared to be of type *WikiArticle*. Pre-existing Cyc relations are used to represent access information. The *topicOfPCW* relation (topic of propositional conceptual work) is used to identify Cyc ontology concept the article represented refers to. Once this link is made, Cyc can answer logical queries involving *WikiArticles* and other Cyc collections and relationships.

Figure 13: Entry in Cyc for *ReferenceExample* article on Jimmy Carter.

4.5 Detecting the Need for New Classes

One key situation faced by any self directed learning process is the need to recognize when new classes need to be created. Once a class node is expanded into its children subclasses, the success of finding a classification in them can be monitored. If none of the children produce a classification then one of three conditions has occurred:

- The parent classification was a false positive at level n-1, and the expansion was a mistake.
- A false negative occurred with one of the children at level n.
- An incomplete cover exists and new class needs to be defined at level n as a child of the parent at level n-1.

The benefit of the search framework is that the performance of each classifier is measured, and this information can be used to estimate the likelihood of needing a new class. The lower the error rate of the classifiers involved, the higher the probability of

having discovered a new class. An estimate of the need for a new case can be derived by estimating the probability the parent classifier is right in its positive classification and if all the children were correct in their negative classification:

$$probability\_of\_new\_class = Parent\_Class\_\Pr ecision \times \prod_{i \in Children\_of\_Parent\_Class} Child\_Negative\_\Pr ecision_i \quad (4.1)$$

$$probability\_of\_new\_class = Parent\_Class\_Accuracy \times \prod_{i \in Children\_of\_Parent\_Class} Child\_Accuracy_i \quad (4.2)$$

4.6 Summary

In this system, a method of traversing the ontology in a way to classify new instances in promising ways was defined. It used breadth-first search over the ontology, using an expansion function defined by a set of classes marked as suitable for text classification. In a system with more resources, all child nodes would be allowed or inference rules used. Eventually, machine learning could be turned on the task of discovering what makes a class good for text classification (but that is for a future work).

CHAPTER 5

KNOWN COMPROMISES AND SYSTEM EVALUATION

5.1 Overview

In this section I consider the performance of the system relative to the information it is able to provide about the topics of Wikipedia articles. First, I identify the compromises and limitation of the data and methods that were used. Addressing these limitations will be the part of any future system development. Having identified the compromises, I examine the performance of the classifiers the system generates and what they say about the Wikipedia, Cyc and the types of information classifiable with the n-gram classifiers used.

5.2 Known Compromises

Several compromises were made relative to an ideal implementation for classifying all Wikipedia articles relative to all Cyc categories:

- Restrictions were placed on the training of the classifiers
- Restrictions were placed on the set of categories used for training classifiers
- Possible limitations of the original data set used

5.2.1 Restriction in the Training Classifiers

The first compromise was in the use of iterative training instead of using the complete corpus. Having identified the positive and negative articles to represent a given Cyc concept, the system adds to the training set of the classifiers by expanding the trained-on-set with those articles that have been mismatched. This process allows statistical

snapshots to be taken at each step and help plot the effects of adding Wikipedia articles to the classifiers. It also promotes generalization and prevents over-specialization. This is important when only a few articles are available for some classes. Iterative learning also reduces the training time and space requirements. The n-gram representations tend to require $O(w^n)$ storage, where w is the number of unique words and n is the n-gram size chosen. Iterative learning increments w until either the minimum performance criteria are reached, or all available documents are used.

The second compromise in training the classifiers is in the definition of the training termination criteria. The termination criteria are:

1) The classifier has reached 90% overall accuracy on the iterative test
2) A maximum of k iterations has occurred (k was chosen to be =8)
3) All of the positive and negative training instances are exhausted

In some cases the criteria would stop the training before maximum classifier performance level is reached. There is therefore a need to explore optimal termination criteria. One possibility would be to terminate training when the difference in the improvement in the classifier performance drops below a certain threshold.

5.2.2 Restrictions on the Trainable Categories

One of the features discovered while examining the data was that some categories form virtual chains in the ontology relative to the classification of set of Wikipedia articles. That is to say several Cyc categories will subsume each other without including different articles. Thus these classes would not be discriminatory relative to the data

set. This can be due to the Cyc ontology making finer grained distinctions than the authors of Wikipedia articles. Exploring such chains by the system would add additional work without providing any additional discriminative power.

This effect can be detected when counts are made of Cyc concepts indexed by articles. One can see several concepts in a subclass hierarchy relationship in Cyc without increasing the number of articles covered which would be expected if additional articles were being covered by higher-level/more general concepts.

Cyc Concept	Number of Articles
#$GeographicalRegion	458
#$City	385
#$LandTopographicalFeature	266
#$Place	254
#$GeographicalThing	247
#$CountrySubsidiary	231
#$County	229
#$IndependentCountry	229
#$Province	229
#$GeographicalAgent	156
#$GeopoliticalEntity	156
#$Municipality	156

Table 2: Separate, sub concepts of *#$GeographicalRegion* have some redundant coverage of articles.

Cyc Concept	Number of Articles
#$SpatialThing	2528
#$Location-Underspecified	997
#$Region-Underspecified	969
#$Landmark-Underspecified	850
#$Boundary-Underspecified	849
#$SpatialThing-Localized	849

Table 3: Separate, sub concepts of *#$SpatialThing* also have some redundant coverage of articles.

These sets of articles are covered by multiple concept definitions which are distinguished as separate concepts by the generalization relationship in the Cyc ontology. This effect prompted the restriction in the number of trainable categories to be a set that provided maximal coverage with minimum redundancy from the viewpoint of the Wikipedia articles used.

5.2.3 Limits of the Initial Data Source

To make the system work, an initial set of seed mappings (preferably as close to the leaves as possible) is necessary to form the positive and negative training sets. The researcher Sudarshan Palliyill at IBM Bangalore India provided through the Cyc Foundation the set of 9000 Wikipedia articles that he manually mapped to Cyc concepts for his own research. Most of these articles were in fact leaves describing specific people, places and things. Each mapping provides the basic information: *Wikipedia article A describes Cyc concept C.*

Since the system's performance is based on the quality of its training set, both the primary benefit and limitation is that the training set is provided by one very dedicated judge. The benefit is consistency, while the limitation is possible disagreement between the judge and other potential rankers.

This problem can be addressed by review of both random samples and hard-to-classify articles. As time progresses additional Wikipedia to Cyc mappings will become available.

5.3 Training and Performance

Performance statistics were collected on the performance of the trained classifiers using the measures described in Section 3.2, Measuring classifier performance. When tested on a hold out sample set of 20 documents the system used only a subset of the original set of 418 allowed classes, creating only those classifiers needed to classify those in the sample. An additional measure, "Depth" in the ontology was estimated by running a query to estimate the distance from the concept to the ontology root concept *#$Thing*. The query run was:

```
(length (WHY-COLLECTIONS-INTERSECT?  <cyc-concept> #$Thing))
```

This query measures the number of steps Cyc used to explain how the <cyc-concept> is related to *#$Thing*.

The gold standard was the original manual classification of the articles relative to the Cyc ontology. For any manually assigned article the set of valid categories in the ontology can be determined by asking Cyc for the generalizations from the assignment point. The accuracy measure given is of the classifier's performance relative to the selected generated training sets.

Cyc Concept	Accuracy	+ F-measure	- F-measure	Trials	Depth
AdultAnimal	100.00	1.00	1.00	1	10
Constellation	100.00	1.00	1.00	1	8
ConstructionArtifact	100.00	0.00	1.00	8	7
ConsumableProduct	100.00	1.00	1.00	1	4
GovernmentEmployee	100.00	1.00	1.00	7	14
OrganismPart	100.00	1.00	1.00	1	8
Politician	100.00	1.00	1.00	1	8
Product	100.00	1.00	1.00	1	2
TacticalTerrainObject	100.00	0.00	1.00	1	8
Workplace	100.00	0.00	1.00	1	9
Artifact	96.15	0.96	0.96	3	6
BiologicalLivingObject	95.65	0.96	0.94	2	7
PerceptualAgent	94.73	0.93	0.95	8	8
Artifact-NonAgentive	94.11	0.95	0.92	1	7
GeopoliticalEntity	93.75	0.00	1.00	1	9
Technology-Artifact	93.54	0.95	0.90	1	5
Agent-NonGeographical	92.59	0.92	0.92	1	7
Agent-Generic	92.00	0.92	0.90	4	6
ConventionalClassificationType	91.66	0.00	0.96	1	3
Roadway	90.90	0.67	0.95	1	2
FirstOrderCollection	90.32	0.85	0.92	1	4
IntelligentAgent	84.38	0.74	0.89	8	8
TimeDependentCollection	79.41	0.36	0.88	8	3
AnimalBodyPart	73.33	0.60	0.80	8	10
Animal	65.21	0.67	0.64	8	9
Artifact-Generic	63.33	0.75	0.26	8	6
TransportFacility	55.55	0.67	0.33	8	2
Place	54.83	0.69	0.13	8	7
GeographicalRegion	53.13	0.48	0.57	8	2
LegalGovernmentOrganization	50.00	0.00	0.67	8	9
AnimalBLO	45.45	0.53	0.36	8	8
HumanOccupationConstruct	40.00	0.57	0.00	8	8
Individual	36.36	0.46	0.22	8	3
Cell	29.16	0.00	0.45	8	8
GeographicalPlace	25.72	0.00	0.41	8	8
Graphic-VisualIBT	24.24	0.00	0.39	8	7
PartiallyTangibleProduct	24.00	0.00	0.39	8	3
LinguisticExpressionType	23.80	0.00	0.38	8	4
SexuallyMature	15.79	0.00	0.27	8	9
PartOfBuilding	15.78	0.20	0.11	8	7
Airport-Physical	13.63	0.00	0.24	8	3
GovernmentRelatedEntity	13.33	0.00	0.24	8	7

Table 4: Trained classifier performance (continued on next page).

(Table continued from previous page.)

Cyc Concept	Accuracy	+ F-measure	- F-measure	Trials	Depth
PathArtifact	11.76	0.21	0.00	8	8
AnimalBodyRegion	10.52	0.19	0.00	8	9
Artifact-Agentive	8.33	0.00	0.15	8	7
Organization	8.33	0.00	0.15	8	8
FictionalCharacter	6.00	0.13	0.00	8	7
Collection	0.00	0.00	0.00	8	2

Table 4: Trained classifier performance.

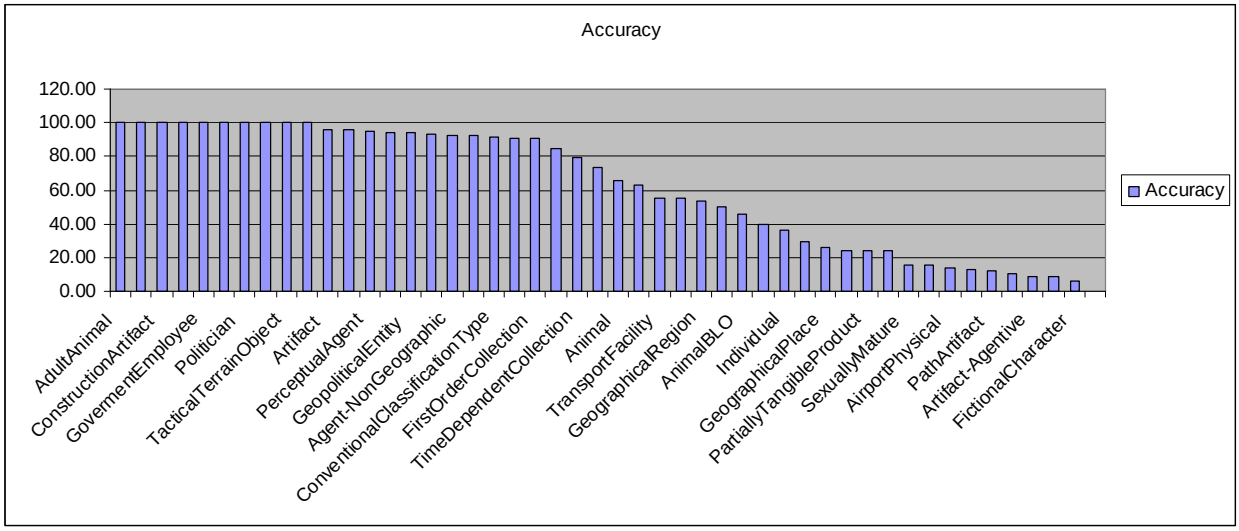
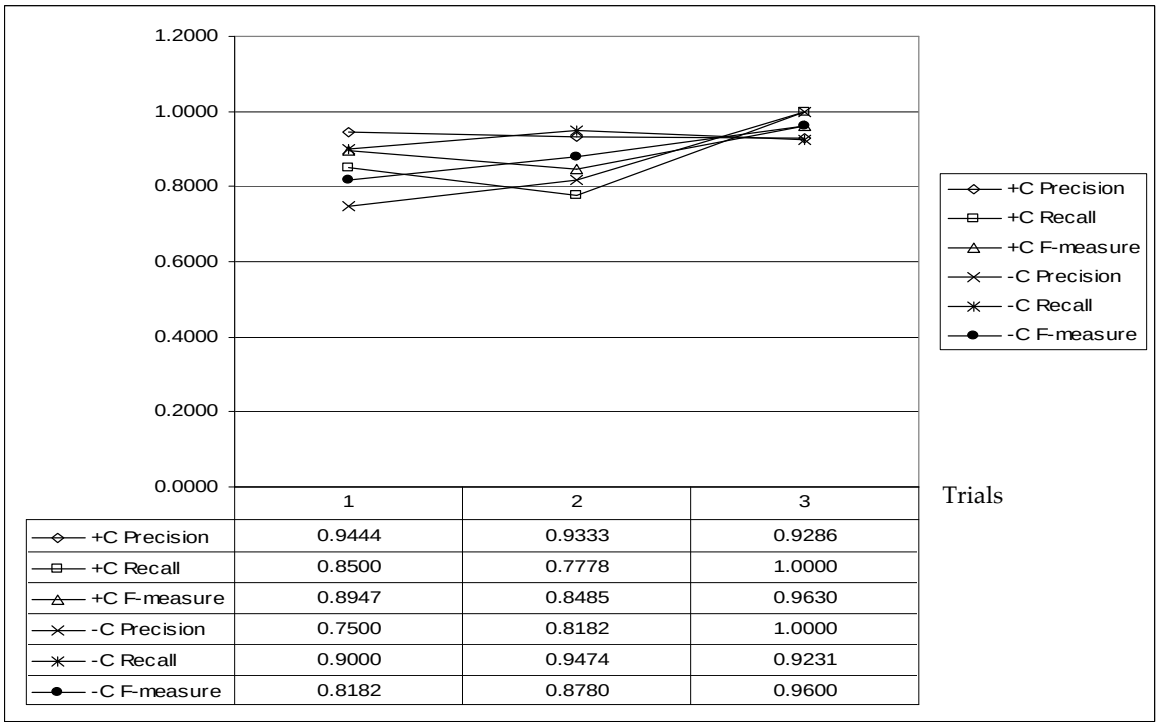


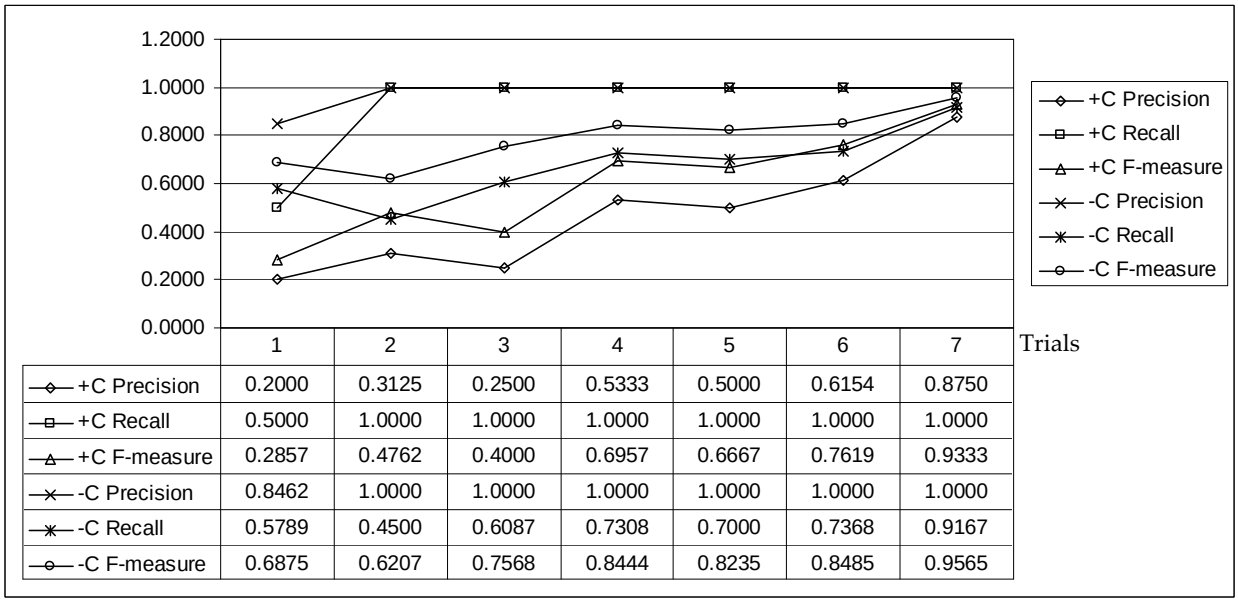
Figure 14: Plot of classifier accuracy by class.

Figure 14 above displays the final accuracy of the 48 classifiers generated. While 50% of those generated have a better than 70% accuracy, 43.7% of the classifiers generated have a better than 90% accuracy. Thus a large number of generated classifiers should have usable accuracy. However, 25% score below 25% on measured accuracy. This points towards the need to explore other classification methods for the identified categories.



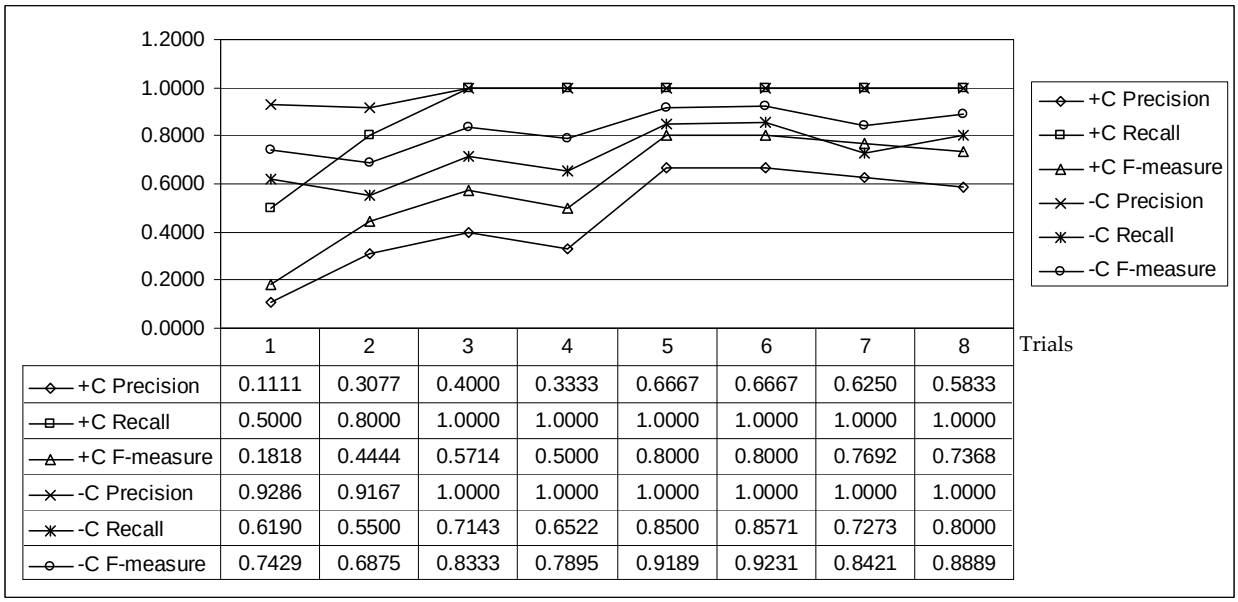
	1	2	3
+C Precision	0.9444	0.9333	0.9286
+C Recall	0.8500	0.7778	1.0000
+C F-measure	0.8947	0.8485	0.9630
-C Precision	0.7500	0.8182	1.0000
-C Recall	0.9000	0.9474	0.9231
-C F-measure	0.8182	0.8780	0.9600

Figure 15: Training of class *Artifact.*



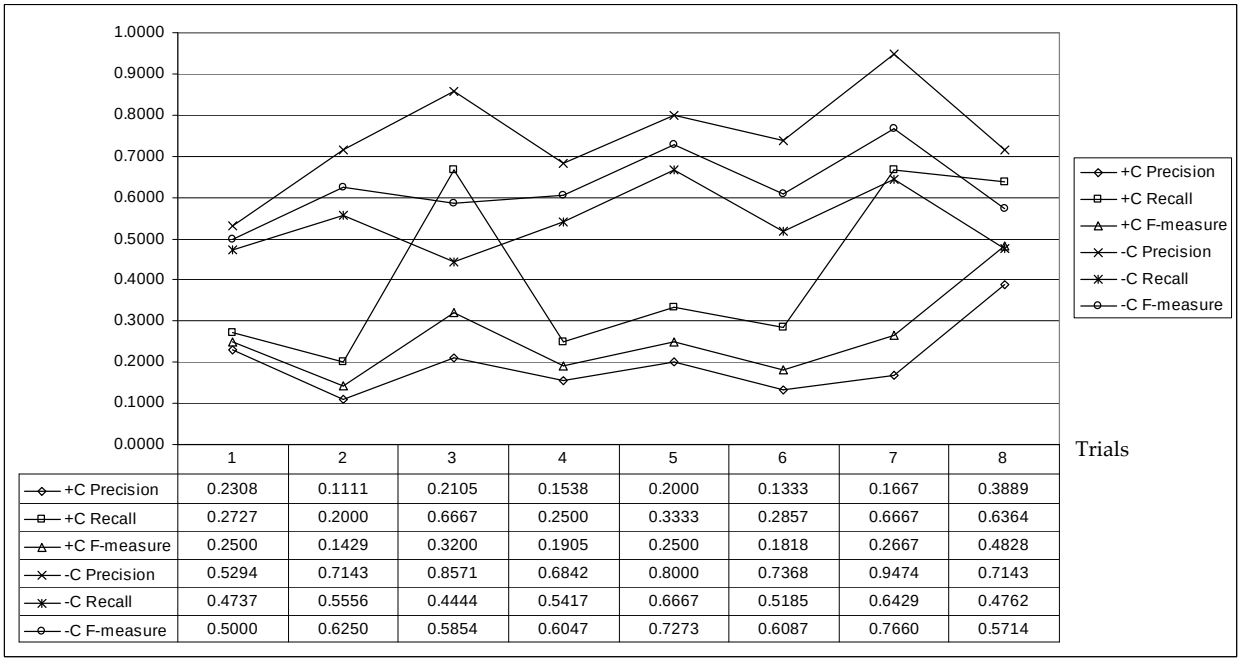
	1	2	3	4	5	6	7
+C Precision	0.2000	0.3125	0.2500	0.5333	0.5000	0.6154	0.8750
+C Recall	0.5000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
+C F-measure	0.2857	0.4762	0.4000	0.6957	0.6667	0.7619	0.9333
-C Precision	0.8462	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
-C Recall	0.5789	0.4500	0.6087	0.7308	0.7000	0.7368	0.9167
-C F-measure	0.6875	0.6207	0.7568	0.8444	0.8235	0.8485	0.9565

Figure 16: Training of class *PerceptualAgent.*



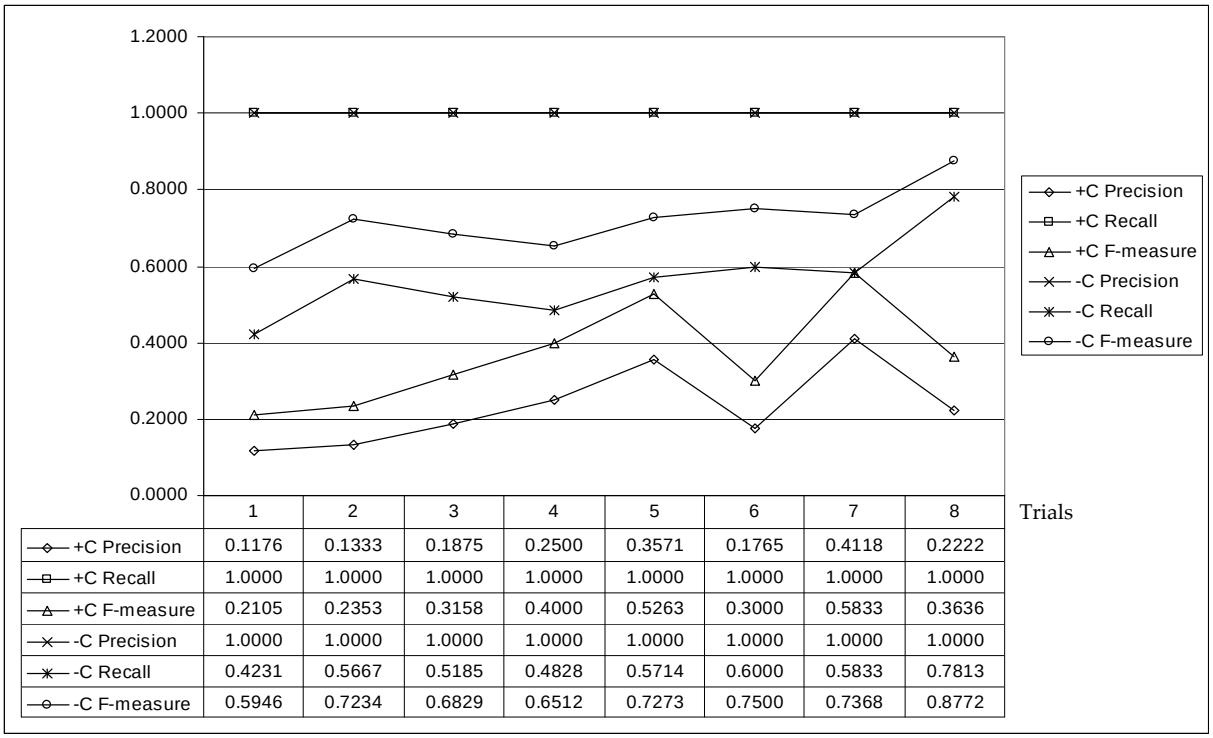
	1	2	3	4	5	6	7	8
Trials								
+C Precision	0.1111	0.3077	0.4000	0.3333	0.6667	0.6667	0.6250	0.5833
+C Recall	0.5000	0.8000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
+C F-measure	0.1818	0.4444	0.5714	0.5000	0.8000	0.8000	0.7692	0.7368
-C Precision	0.9286	0.9167	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
-C Recall	0.6190	0.5500	0.7143	0.6522	0.8500	0.8571	0.7273	0.8000
-C F-measure	0.7429	0.6875	0.8333	0.7895	0.9189	0.9231	0.8421	0.8889

Figure 17: Training of class *IntelligentAgent*.



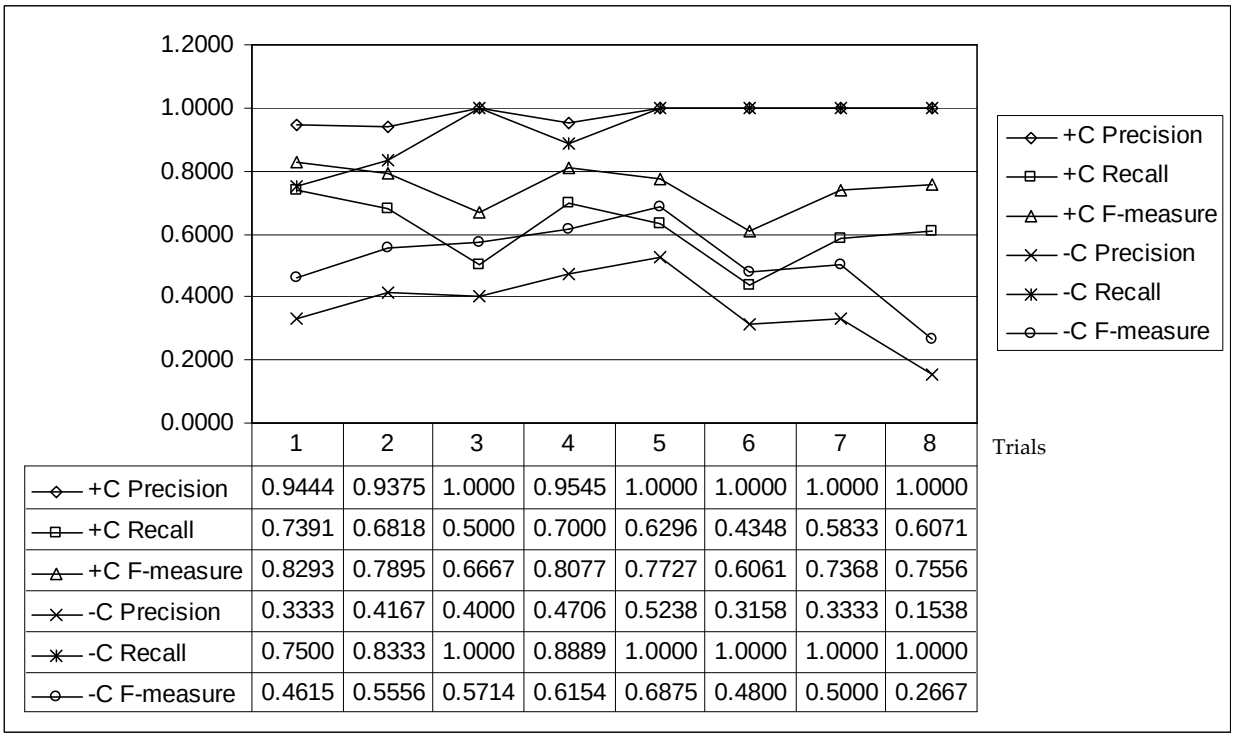
	1	2	3	4	5	6	7	8
Trials								
+C Precision	0.2308	0.1111	0.2105	0.1538	0.2000	0.1333	0.1667	0.3889
+C Recall	0.2727	0.2000	0.6667	0.2500	0.3333	0.2857	0.6667	0.6364
+C F-measure	0.2500	0.1429	0.3200	0.1905	0.2500	0.1818	0.2667	0.4828
-C Precision	0.5294	0.7143	0.8571	0.6842	0.8000	0.7368	0.9474	0.7143
-C Recall	0.4737	0.5556	0.4444	0.5417	0.6667	0.5185	0.6429	0.4762
-C F-measure	0.5000	0.6250	0.5854	0.6047	0.7273	0.6087	0.7660	0.5714

Figure 18: Training of class *GeographicalRegion*.



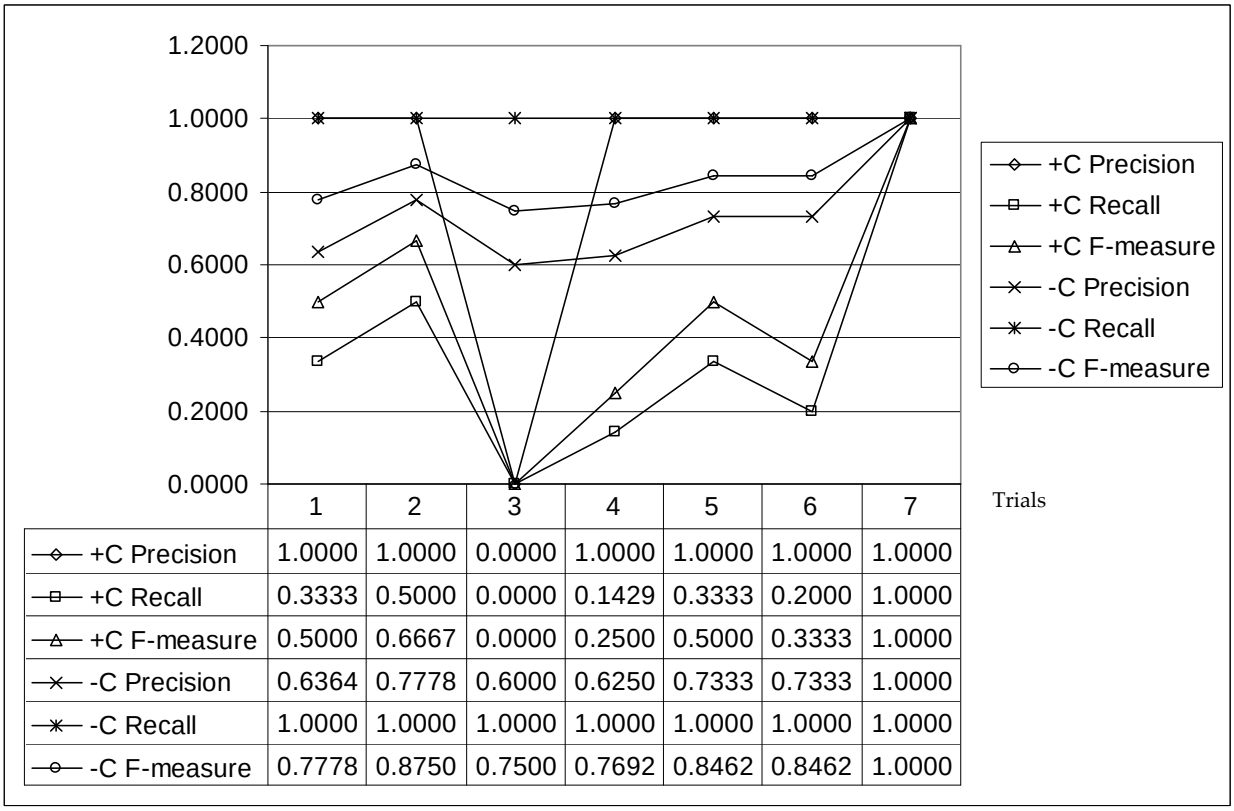
	1	2	3	4	5	6	7	8
Trials								
+C Precision	0.1176	0.1333	0.1875	0.2500	0.3571	0.1765	0.4118	0.2222
+C Recall	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
+C F-measure	0.2105	0.2353	0.3158	0.4000	0.5263	0.3000	0.5833	0.3636
-C Precision	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
-C Recall	0.4231	0.5667	0.5185	0.4828	0.5714	0.6000	0.5833	0.7813
-C F-measure	0.5946	0.7234	0.6829	0.6512	0.7273	0.7500	0.7368	0.8772

Figure 19: Training of class *TimeDependentCollection.*



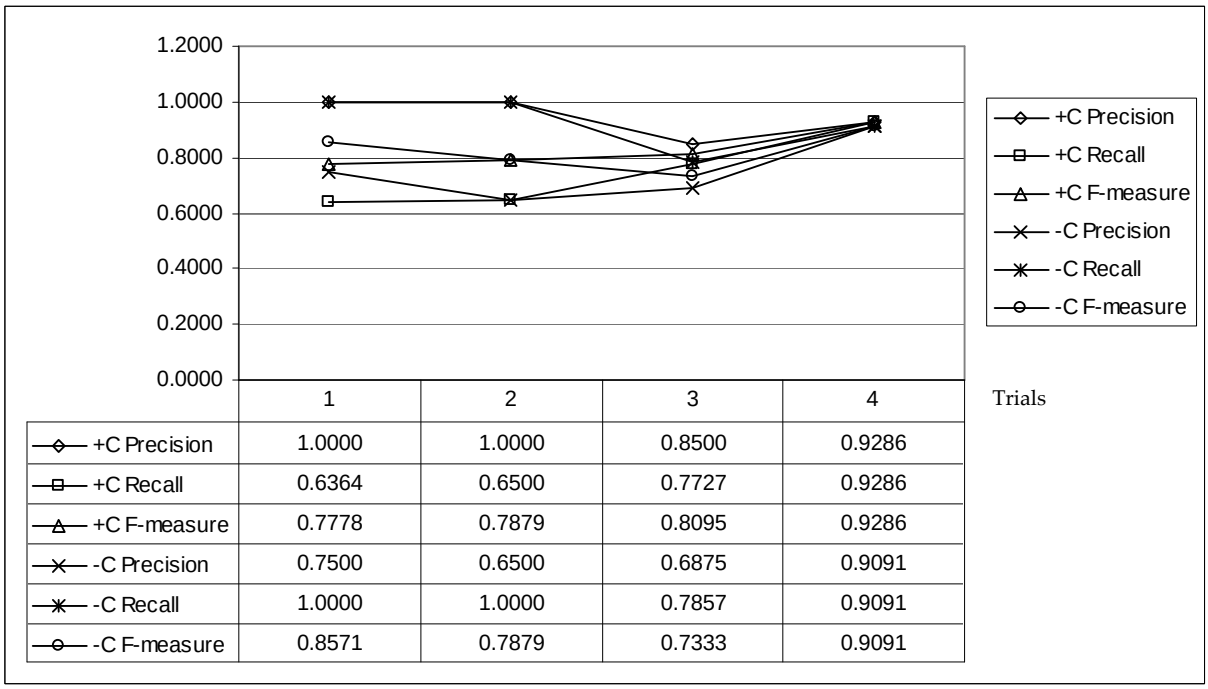
	1	2	3	4	5	6	7	8
Trials								
+C Precision	0.9444	0.9375	1.0000	0.9545	1.0000	1.0000	1.0000	1.0000
+C Recall	0.7391	0.6818	0.5000	0.7000	0.6296	0.4348	0.5833	0.6071
+C F-measure	0.8293	0.7895	0.6667	0.8077	0.7727	0.6061	0.7368	0.7556
-C Precision	0.3333	0.4167	0.4000	0.4706	0.5238	0.3158	0.3333	0.1538
-C Recall	0.7500	0.8333	1.0000	0.8889	1.0000	1.0000	1.0000	1.0000
-C F-measure	0.4615	0.5556	0.5714	0.6154	0.6875	0.4800	0.5000	0.2667

Figure 20: Training of class *Artifact-Generic.*



	1	2	3	4	5	6	7
+C Precision	1.0000	1.0000	0.0000	1.0000	1.0000	1.0000	1.0000
+C Recall	0.3333	0.5000	0.0000	0.1429	0.3333	0.2000	1.0000
+C F-measure	0.5000	0.6667	0.0000	0.2500	0.5000	0.3333	1.0000
-C Precision	0.6364	0.7778	0.6000	0.6250	0.7333	0.7333	1.0000
-C Recall	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
-C F-measure	0.7778	0.8750	0.7500	0.7692	0.8462	0.8462	1.0000

Figure 21: Training of *GovernmentEmployee.*



	1	2	3	4
+C Precision	1.0000	1.0000	0.8500	0.9286
+C Recall	0.6364	0.6500	0.7727	0.9286
+C F-measure	0.7778	0.7879	0.8095	0.9286
-C Precision	0.7500	0.6500	0.6875	0.9091
-C Recall	1.0000	1.0000	0.7857	0.9091
-C F-measure	0.8571	0.7879	0.7333	0.9091

Figure 22: Training of *Agent-Generic.*

5.4 Summary

Given the identified limitations, the system was able to gather useful information on both the possibility and limitations of the overall approach. First, the data collected showed that for a subset of the ontology the n-gram based language models could be trained to discriminate in-category articles from out-of-category articles. Approximately 50% of the classifiers had greater than 70% accuracy and 43% had greater than 90% accuracy. The general search procedure only generated the classifiers required to classify the randomly selected test cases. The collected information showed that 25% of the classifiers have less than 25% accuracy. This highlights the need to explore additional classification methods. These classifiers can then be integrated in the overall framework (which would be a promising direction for future work).

CHAPTER 6

RELATED WORK

Several approaches are used currently to classify the content of documents or gather object relationships information from them. This is a very brief list of related work in text classification, gathering relationship information from both text and humans, and some self directed search processes.

6.1 N-gram Text Classification

N-gram based text classification has a long history of use. Cavnar [Cavnar et al., 1994] discussed the use of n-grams for classifying USENET postings by location in topic hierarchy. As discussed in Section 3.6 [Mladenic et al., 2004] used similar text classification method to identify the location of new web pages in a Web directory like Yahoo. And, as outlined in Section 3.4, Analogy to other text classification methods, E-mail spam detection and filtering is one area where n-gram based language models are used on a daily basis. From a purely textual viewpoint spam is a fuzzy concept. The goal of a trainable spam classification is: given example E-mails classified by the user as either "spam" (non-user relevant) or "ham" (user-relevant), generate a text based classifier that can discriminate between the two on new emails. Despite spam being often crafted to appear to be potentially user relevant, classifier accuracy of 99.9% accuracy have been reported [Yerazunis, 2003; Yerazunis, 2004; Yerazunis et al., 2005]. The system tested in this thesis using similar methods to those used in e-mail spam classification achieved reasonable performance (>90% accuracy) in over 40% of the category classifiers generated.

6.2 Knowledge Extraction by Reading

Different methods have been attempted to directly extract relationships by parsing texts in various ways. One method utilizes existing knowledge of a known relationship (like "Austin is the capital of Texas") and finds either surface text patterns or patterns in the output of various parsers. Examples of systems that detect surface patterns include VERBOCEAN [Chklovski et al., 2004] and some methods used by pattern based answer extraction systems [Hovy et al., 2002a; Hovy et al., 2002b; Lin et al., 2001; Ravichandran et al., 2002; Etzioni et al., 2004]. Another method detects characteristic patterns in the output of parsers [Schubert, 2002; Schubert, 2003; Fleischman et al., 2003] instead of surface patterns.

Several researchers [Duthie et al., 2002; Molla-Aliod et al., 2002; Zhang et al., 2001] utilize this approach to extract relationships from the dependency output of the Link Grammar Parser system [Sleator et al., 1991].

Another method of extracting relationships is via collecting various co-occurrence statistics over a large corpus and using those to generate extraction patterns. The Terascale extraction system is an example of this approach [Pantel et al., 2004a; Pantel et al., 2004b]. Given groups of words making up a given class, the system is able to identify both co-occurrence and surface text patterns that indicate new phrases are also members of that class in an efficient manner.

All of these methods look at the detailed relations between words. This is in contrast with the system tested in this thesis, which utilizes the statistical properties of the entire

document to determine at each step whether or not a particular logical statement should be made. The presence of certain surface patterns, and by proxy syntactic patterns, is captured by the n-gram construction process.

There is another way of stating the difference between the methods. The detailed relationship methods focus on the data found in one n-gram such as "Austin is capital of Texas," while the statistical approach is able to look at "is capital of," "the legislature is located," "the governor mansion," "became the capital," etc., when making its determination that an article on "Austin" is about a "Capital."

6.3 Expanding the Knowledge in Ontologies Directly

Various methods have been used to increase the body of knowledge of systems like Cyc. One has been to ask volunteers for fragments of commonsense information. This was seen in the continuing OpenMind project [Singh et al., 2002]. Another includes interactive knowledge elicitation sessions with domain experts [Matthews, 2003; Panton et al., 2002; Witbrock et al., 2003]. While lowering the barrier to generating knowledge, these approaches maintain a tight human interaction coupling. Humans are the source for each fact entered into the system. This is in contrast to system tested in the thesis, which utilizes the initial human classifications and applies it to new text.

Using focused pattern based search engine queries [Matuszek et al., 2005; Witbrock et al., 2005] to both generate and validate specific relationship hypothesis is also being explored.

6.4 Agenda-Based Search

The system developed implements a breath first search across the set of Cyc categories in a manner similar to an agenda system. A priority task queue sorted by an interest evaluation function has been used before for self-directed search, and in systems related to Cyc. The Artificial Mathematician (AM) system [Lenat, 1976] and Eurisko [Lenat, 1984] both used agenda mechanisms to rank the progression of work. Work on these two systems influenced the eventual development of Cyc. Two follow-on discovery systems, Cyrano [Haase, 1986] and HAMB [Livingston, 2001], also used the agenda mechanism framework. The primary difference between the discovery systems and this one is discovery systems often generate new hypothetical formula that require analysis, while in this system the search space is determined by the Cyc ontology, the single classifier used and the set of articles. However, future systems would be more like agenda driven discovery systems by allowing the creation of new categories, the provision of additional classifier types and expanding the article set to the open internet.

6.5 Summary

In summary, many methods exist to gather information about the objects in the real world through statistical analysis of text, analysis of the closer relationships between elements of text, or focused knowledge elicitation or probing. No one method is perfect, each having its own assumption about the availability of resources.

CHAPTER 7

CONCLUSIONS

7.1 Summary

Automatic mapping of Wikipedia articles to an ontology like Cyc is possible, given enough seed material. I was able to demonstrate the automatic creation of language models using a traditional search method guided by a large-scale ontology. The ontology not only provided a list or classes to be created, but also organized the partitioning of training instances. The overall algorithm should be general enough to guide the creation of classifiers in other domains.

7.2 Future Work

The system provides several areas for future work:

- Modify the framework to allow the use of additional classifiers, and provide the features necessary to reason about their use. This would entail reflecting the classifier accuracy information back into the KB, and providing additional rules to select the best classifier on a per-class basis during the classifier selection and creation process.
- Given the "whole text" approach taken, the system could be trained to provide context to Cycorp's shallow parsing system, which requires a domain category as its initial input.
- Extraction and association of key phrases with topic concepts. The language model is based on word n-grams. The highly relevant n-grams could be extracted and used for additional search keys when searching for articles on the web or Wikipedia.
- Extend the system to work with other descriptive texts other than just Wikipedia articles.

- Improve performance via better speed and better compression of the classifiers. Some performance gain could be achieved by conversion to C/C++ for the classifier code.
- Use Cyc concepts in the indexing process. Do a denotational-map of the article (what phrases ambiguously map to which concepts), and then add the terms as unigrams for article discrimination.
- Extend the overall process to use Weka and other machine learning systems.
- Extend the method to cover classes other than text. For instance, replace the text classifiers with image classifiers.
- Bootstrapping by merging articles classified with high confidence into future training sets.
- Convert the system to run as a Condor parallel batch system process so the system can be applied to all available Wikipedia articles and all Cyc classes.
- Remove some of the known restrictions when using a large distributed computing environment.

7.3 New Technology Implemented

Several new things were developed in this thesis. Cyc was used to control the generation of language model based classifiers that can be applied to Wikipedia articles to make statements about the nature of their topic. The framework is fairly simple yet extendable and can include other classification techniques.

7.4 Research Results

I was able to determine that the accuracy of the type of language models used varies greatly depending on the Cyc concept being trained on Wikipedia articles. For example, fictional characters are described in a way that real people are described. This is an area where other analysis methods would be relevant. However, a significant

number of concepts can be learned to a usable accuracy level. Approximately 50% of the classifiers had greater than 70% accuracy and 43% had greater than 90% accuracy.

Another fact discovered is that the Cyc ontology has a finer grain of distinction that the authors of the Wikipedia articles examined. Cyc's core was written by authors trained in logic seeking to provide an unambiguous, formal description. In contrast, the Wikipedia authors are volunteers using natural language. Cyc is providing an extra level of information or distinctions missing in the articles. Possible redundancy might exist in the KB due to multiple knowledge engineers tasked in different domains naming the same concept with different terms.

7.5 Conclusions

These results support the possibility of self-directed creation of classifiers using an ontology as an organizing mechanism. In particular, it shows that broad large-scale ontologies like Cyc allow the classification of items described by Wikipedia articles in a semi-autonomous way. The primary requirement is an initial set of instances associated at the lowest levels of the ontology.

While only a proof-of-concept prototype, the system shows the general usefulness of the overall framework. Additional classification mechanisms can be utilized, and the overall system can be applied to elements other than text. As larger sources like the Wikipedia and indeed the Web in general become available and larger ontologies like Cyc or the Semantic Web become available, the ability to map one into the other

automatically and to extend them when necessary will become increasingly important. This research provides a step in that direction.

APPENDIX A

EXAMPLE CLASSIFIER RUNS

BiologicalLivingObject

```
------------------------------------------------------------
Quiz :  #$BiologicalLivingObject 1
 QPS:1  QPR: 15  QPC:15
 QNS:0.727272727272727  QNR: 8  QNC:11
 QCS:0.884615384615385  QCR: 23  QCC:26
 === Detailed Accuracy By Class ===
 TP Rate   FP Rate    Precision   Recall   F-Measure   Class
  1.0000    0.0000     1.0000     1.0000    0.9091     + #$BiologicalLivingObject
  0.7273    0.2727     0.7273     0.7273    0.8421     - #$BiologicalLivingObject
 === Confusion Matrix: #$BiologicalLivingObject 1  ===
   +Class      -Class     <-----Classified as
 TP: 0015    FP: 0000   + #$BiologicalLivingObject
 FN: 0003    TN: 0008   - #$BiologicalLivingObject
 ==========TEST ERROR==========
 Correctly Classified Instances   2388.4615384615385 %
 Incorrectly Classified Instances  311.5384615384615 %
 Total Number of Instances       26
------------------------------------------------------------
------------------------------------------------------------
Quiz :  #$BiologicalLivingObject 2
 QPS:1  QPR: 10  QPC:10
 QNS:0.857142857142857  QNR: 6  QNC:7
 QCS:0.941176470588235  QCR: 16  QCC:17
 === Detailed Accuracy By Class ===
 TP Rate   FP Rate    Precision   Recall   F-Measure   Class
  1.0000    0.0000     1.0000     1.0000    0.9524     + #$BiologicalLivingObject
  0.8571    0.1429     0.8571     0.8571    0.9231     - #$BiologicalLivingObject
 === Confusion Matrix: #$BiologicalLivingObject 2  ===
   +Class      -Class     <-----Classified as
 TP: 0010    FP: 0000   + #$BiologicalLivingObject
 FN: 0001    TN: 0006   - #$BiologicalLivingObject
 ==========TEST ERROR==========
 Correctly Classified Instances   1694.1176470588235 %
 Incorrectly Classified Instances  15.88235294117647 %
 Total Number of Instances       17
------------------------------------------------------------
TestArticle:Ronald Reagan   TargetConcept :#$BiologicalLivingObject  classDecision = T confidence:26669.3372020782
```

Place

Quiz : #$Place 1
QPS:0.0833333333333333 QPR: 1 QPC:12
QNS:1 QNR: 30 QNC:30
QCS:0.738095238095238 QCR: 31 QCC:42
=== Detailed Accuracy By Class ===

TP Rate	FP Rate	Precision	Recall	F-Measure	Class
0.0833	0.9167	0.0833	0.0833	0.1538	+ #$Place
1.0000	0.0000	1.0000	1.0000	0.8451	- #$Place

=== Confusion Matrix: #$Place 1 ===

+Class	-Class	<-----Classified as
TP: 0001	FP: 0011	+ #$Place
FN: 0000	TN: 0030	- #$Place

==========TEST ERROR==========
Correctly Classified Instances 3173.8095238095238 %
Incorrectly Classified Instances 1126.1904761904762 %
Total Number of Instances 42

Quiz : #$Place 2
QPS:0.181818181818182 QPR: 4 QPC:22
QNS:0.947368421052632 QNR: 18 QNC:19
QCS:0.536585365853659 QCR: 22 QCC:41
=== Detailed Accuracy By Class ===

TP Rate	FP Rate	Precision	Recall	F-Measure	Class
0.1818	0.8182	0.1818	0.1818	0.2963	+ #$Place
0.9474	0.0526	0.9474	0.9474	0.6545	- #$Place

=== Confusion Matrix: #$Place 2 ===

+Class	-Class	<-----Classified as
TP: 0004	FP: 0018	+ #$Place
FN: 0001	TN: 0018	- #$Place

==========TEST ERROR==========
Correctly Classified Instances 2253.6585365853659 %
Incorrectly Classified Instances 1946.3414634146341 %
Total Number of Instances 41

Quiz : #$Place 3
QPS:0.5 QPR: 4 QPC:8
QNS:1 QNR: 13 QNC:13
QCS:0.80952380952381 QCR: 17 QCC:21
=== Detailed Accuracy By Class ===

TP Rate	FP Rate	Precision	Recall	F-Measure	Class
0.5000	0.5000	0.5000	0.5000	0.6667	+ #$Place
1.0000	0.0000	1.0000	1.0000	0.8667	- #$Place

=== Confusion Matrix: #$Place 3 ===

+Class	-Class	<-----Classified as
TP: 0004	FP: 0004	+ #$Place

FN: 0000 TN: 0013 - #$Place

==========TEST ERROR==========

Correctly Classified Instances 1780.9523809523809 %

Incorrectly Classified Instances 419.047619047619 %

Total Number of Instances 21

Quiz : #$Place 4

QPS:0.555555555555556 QPR: 10 QPC:18

QNS:1 QNR: 25 QNC:25

QCS:0.813953488372093 QCR: 35 QCC:43

=== Detailed Accuracy By Class ===

TP Rate	FP Rate	Precision	Recall	F-Measure	Class
0.5556	0.4444	0.5556	0.5556	0.7143	+ #$Place
1.0000	0.0000	1.0000	1.0000	0.8621	- #$Place

=== Confusion Matrix: #$Place 4 ===

+Class -Class <-----Classified as

TP: 0010 FP: 0008 + #$Place

FN: 0000 TN: 0025 - #$Place

==========TEST ERROR==========

Correctly Classified Instances 3581.3953488372093 %

Incorrectly Classified Instances 818.6046511627907 %

Total Number of Instances 43

Quiz : #$Place 5

QPS:0.444444444444444 QPR: 8 QPC:18

QNS:1 QNR: 7 QNC:7

QCS:0.6 QCR: 15 QCC:25

=== Detailed Accuracy By Class ===

TP Rate	FP Rate	Precision	Recall	F-Measure	Class
0.4444	0.5556	0.4444	0.4444	0.6154	+ #$Place
1.0000	0.0000	1.0000	1.0000	0.5833	- #$Place

=== Confusion Matrix: #$Place 5 ===

+Class -Class <-----Classified as

TP: 0008 FP: 0010 + #$Place

FN: 0000 TN: 0007 - #$Place

==========TEST ERROR==========

Correctly Classified Instances 1560 %

Inorrectly Classified Instances 1040 %

Total Number of Instances 25

Quiz : #$Place 6

QPS:0.523809523809524 QPR: 11 QPC:21

QNS:1 QNR: 15 QNC:15

QCS:0.722222222222222 QCR: 26 QCC:36

=== Detailed Accuracy By Class ===

TP Rate	FP Rate	Precision	Recall	F-Measure	Class
0.5238	0.4762	0.5238	0.5238	0.6875	+ #$Place

```
  1.0000    0.0000    1.0000    1.0000    0.7500    - #$Place
 === Confusion Matrix: #$Place 6 ===
   +Class      -Class      <-----Classified as
 TP: 0011    FP: 0010    + #$Place
 FN: 0000    TN: 0015    - #$Place
 ==========TEST ERROR==========
 Correctly Classified Instances   2672.2222222222222 %
 Inorrectly Classified Instances  1027.7777777777778 %
 Total Number of Instances        36
-----------------------------------------------------------------
-----------------------------------------------------------------
Quiz :  #$Place 7
 QPS:0.777777777777778  QPR: 14  QPC:18
 QNS:1  QNR: 14  QNC:14
 QCS:0.875  QCR: 28  QCC:32
 === Detailed Accuracy By Class ===
 TP Rate   FP Rate    Precision    Recall   F-Measure    Class
  0.7778    0.2222    0.7778    0.7778    0.8750    + #$Place
  1.0000    0.0000    1.0000    1.0000    0.8750    - #$Place
 === Confusion Matrix: #$Place 7 ===
   +Class      -Class      <-----Classified as
 TP: 0014    FP: 0004    + #$Place
 FN: 0000    TN: 0014    - #$Place
 ==========TEST ERROR==========
 Correctly Classified Instances   2887.5 %
 Incorrectly Classified Instances  412.5 %
 Total Number of Instances        32
-----------------------------------------------------------------
-----------------------------------------------------------------
Quiz :  #$Place 8
 QPS:0.8  QPR: 8  QPC:10
 QNS:1  QNR: 22  QNC:22
 QCS:0.9375  QCR: 30  QCC:32
 === Detailed Accuracy By Class ===
 TP Rate   FP Rate    Precision    Recall   F-Measure    Class
  0.8000    0.2000    0.8000    0.8000    0.8889    + #$Place
  1.0000    0.0000    1.0000    1.0000    0.9565    - #$Place
 === Confusion Matrix: #$Place 8 ===
   +Class      -Class      <-----Classified as
 TP: 0008    FP: 0002    + #$Place
 FN: 0000    TN: 0022    - #$Place
 ==========TEST ERROR==========
 Correctly Classified Instances   3093.75 %
 Incorrectly Classified Instances  26.25 %
 Total Number of Instances        32
-----------------------------------------------------------------
TestArticle:Ronald Reagan   TargetConcept :#$Place  classDecision = F confidence:21068.409072867
```

Product

```
------------------------------------------------------------
Quiz :  #$Product 1
 QPS:1  QPR: 11  QPC:11
 QNS:1  QNR: 7  QNC:7
 QCS:1  QCR: 18  QCC:18
 === Detailed Accuracy By Class ===
 TP Rate   FP Rate    Precision   Recall   F-Measure   Class
  1.0000    0.0000    1.0000     1.0000    1.0000     + #$Product
  1.0000    0.0000    1.0000     1.0000    1.0000     - #$Product
 === Confusion Matrix: #$Product 1  ===
   +Class     -Class     <-----Classified as
 TP: 0011    FP: 0000   + #$Product
 FN: 0000    TN: 0007   - #$Product
 ==========TEST ERROR==========
 Correctly Classified Instances   18100 %
 Incorrectly Classified Instances  00 %
 Total Number of Instances       18
------------------------------------------------------------
TestArticle:Ronald Reagan   TargetConcept :#$Product  classDecision = F confidence:11707.9196211729
```

TimeInterval

```
------------------------------------------------------------
Quiz :  #$TimeInterval 1
 QPS:1  QPR: 1  QPC:1
 QNS:0  QNR: 0  QNC:13
 QCS:0.0714285714285714  QCR: 1  QCC:14
 === Detailed Accuracy By Class ===
 TP Rate   FP Rate     Precision        Recall   F-Measure        Class
  1.0000    0.0000      1.0000        1.0000     0.1333      + #$TimeInterval
  0.0000    1.0000      0.0000        0.0000     0.0000      - #$TimeInterval
 === Confusion Matrix: #$TimeInterval 1  ===
   +Class       -Class       <-----Classified as
TP: 0001      FP: 0000     + #$TimeInterval
FN: 0013      TN: 0000     - #$TimeInterval
 ==========TEST ERROR=============
 Correctly Classified Instances   17.14285714285714 %
 Inorrectly Classified Instances  1392.8571428571429 %
 Total Number of Instances        14
------------------------------------------------------------
------------------------------------------------------------
Quiz :  #$TimeInterval 2
 QPS:1  QPR: 4  QPC:4
 QNS:0  QNR: 0  QNC:14
 QCS:0.222222222222222  QCR: 4  QCC:18
 === Detailed Accuracy By Class ===
 TP Rate   FP Rate     Precision        Recall   F-Measure        Class
  1.0000    0.0000      1.0000        1.0000     0.3636      + #$TimeInterval
  0.0000    1.0000      0.0000        0.0000     0.0000      - #$TimeInterval
 === Confusion Matrix: #$TimeInterval 2  ===
   +Class       -Class       <-----Classified as
TP: 0004      FP: 0000     + #$TimeInterval
FN: 0014      TN: 0000     - #$TimeInterval
 ==========TEST ERROR=============
 Correctly Classified Instances   422.2222222222222 %
 Inorrectly Classified Instances  1477.7777777777778 %
 Total Number of Instances        18
------------------------------------------------------------
------------------------------------------------------------
Quiz :  #$TimeInterval 3
 QPS:1  QPR: 1  QPC:1
 QNS:0  QNR: 0  QNC:11
 QCS:0.0833333333333333  QCR: 1  QCC:12
 === Detailed Accuracy By Class ===
 TP Rate   FP Rate     Precision        Recall   F-Measure        Class
  1.0000    0.0000      1.0000        1.0000     0.1538      + #$TimeInterval
  0.0000    1.0000      0.0000        0.0000     0.0000      - #$TimeInterval
 === Confusion Matrix: #$TimeInterval 3  ===
   +Class       -Class       <-----Classified as
TP: 0001      FP: 0000     + #$TimeInterval
```

```
 FN: 0011     TN: 0000     - #$TimeInterval
  ==========TEST ERROR=============
  Correctly Classified Instances   18.33333333333333 %
  Incorrectly Classified Instances  1191.6666666666667 %
  Total Number of Instances        12
-------------------------------------------------------------
-------------------------------------------------------------
Quiz :  #$TimeInterval 4
  QPS:1  QPR: 1  QPC:1
  QNS:0  QNR: 0  QNC:14
  QCS:0.0666666666666667  QCR: 1  QCC:15
  === Detailed Accuracy By Class ===
  TP Rate   FP Rate      Precision         Recall    F-Measure         Class
   1.0000     0.0000       1.0000         1.0000      0.1250       + #$TimeInterval
   0.0000     1.0000       0.0000         0.0000      0.0000       - #$TimeInterval
 === Confusion Matrix: #$TimeInterval 4  ===
    +Class       -Class       <-----Classified as
 TP: 0001      FP: 0000     + #$TimeInterval
 FN: 0014      TN: 0000     - #$TimeInterval
  ==========TEST ERROR=============
  Correctly Classified Instances   16.66666666666667 %
  Incorrectly Classified Instances  1493.3333333333333 %
  Total Number of Instances        15
-------------------------------------------------------------
-------------------------------------------------------------
Quiz :  #$TimeInterval 5
  QPS:1  QPR: 2  QPC:2
  QNS:0  QNR: 0  QNC:15
  QCS:0.117647058823529  QCR: 2  QCC:17
=== Detailed Accuracy By Class ===
  TP Rate   FP Rate      Precision         Recall    F-Measure         Class
   1.0000     0.0000       1.0000         1.0000      0.2105       + #$TimeInterval
   0.0000     1.0000       0.0000         0.0000      0.0000       - #$TimeInterval
 === Confusion Matrix: #$TimeInterval 5  ===
    +Class       -Class       <-----Classified as
 TP: 0002      FP: 0000     + #$TimeInterval
 FN: 0015      TN: 0000     - #$TimeInterval
  ==========TEST ERROR=============
  Correctly Classified Instances   211.7647058823529 %
  Inorrectly Classified Instances  1588.2352941176471 %
  Total Number of Instances        17
-------------------------------------------------------------
-------------------------------------------------------------
Quiz :  #$TimeInterval 6
  QPS:1  QPR: 2  QPC:2
  QNS:0  QNR: 0  QNC:13
  QCS:0.133333333333333  QCR: 2  QCC:15
  === Detailed Accuracy By Class ===
  TP Rate   FP Rate      Precision         Recall    F-Measure         Class
   1.0000     0.0000       1.0000         1.0000      0.2353       + #$TimeInterval
```

```
  0.0000    1.0000    0.0000    0.0000    0.0000    - #$TimeInterval
 === Confusion Matrix: #$TimeInterval 6 ===
   +Class      -Class      <-----Classified as
 TP: 0002    FP: 0000    + #$TimeInterval
 FN: 0013    TN: 0000    - #$TimeInterval
  ==========TEST ERROR============
  Correctly Classified Instances   213.3333333333333 %
  Incorrectly Classified Instances  1386.6666666666667 %
  Total Number of Instances       15
-----------------------------------------------------------------
-----------------------------------------------------------------
Quiz :  #$TimeInterval 7
  QPS:1  QPR: 0  QPC:0
  QNS:0  QNR: 0  QNC:23
  QCS:0  QCR: 0  QCC:23
  === Detailed Accuracy By Class ===
  TP Rate   FP Rate    Precision    Recall   F-Measure    Class
   1.0000    0.0000    0.0000    1.0000    0.0000    + #$TimeInterval
   0.0000    1.0000    0.0000    0.0000    0.0000    - #$TimeInterval
 === Confusion Matrix: #$TimeInterval 7 ===
   +Class      -Class      <-----Classified as
 TP: 0000    FP: 0000    + #$TimeInterval
 FN: 0023    TN: 0000    - #$TimeInterval
  ==========TEST ERROR============
  Correctly Classified Instances   00 %
  Incorrectly Classified Instances  23100 %
  Total Number of Instances       23
-----------------------------------------------------------------
-----------------------------------------------------------------
Quiz :  #$TimeInterval 8
  QPS:1  QPR: 0  QPC:0
  QNS:0  QNR: 0  QNC:16
  QCS:0  QCR: 0  QCC:16
  === Detailed Accuracy By Class ===
  TP Rate   FP Rate    Precision    Recall   F-Measure    Class
   1.0000    0.0000    0.0000    1.0000    0.0000    + #$TimeInterval
   0.0000    1.0000    0.0000    0.0000    0.0000    - #$TimeInterval
 === Confusion Matrix: #$TimeInterval 8 ===
   +Class      -Class      <-----Classified as
 TP: 0000    FP: 0000    + #$TimeInterval
 FN: 0016    TN: 0000    - #$TimeInterval
  ==========TEST ERROR============
  Correctly Classified Instances   00 %
  Incorrectly Classified Instances  16100 %
  Total Number of Instances       16
-----------------------------------------------------------------
TestArticle:Ronald Reagan   TargetConcept :#$TimeInterval  classDecision = T confidence:97966.0004581333
```

NaturalThing

Quiz : #$NaturalThing 1
QPS:1 QPR: 13 QPC:13
QNS:0.888888888888889 QNR: 16 QNC:18
QCS:0.935483870967742 QCR: 29 QCC:31
=== Detailed Accuracy By Class ===

TP Rate	FP Rate	Precision	Recall	F-Measure	Class
1.0000	0.0000	1.0000	1.0000	0.9286	+ #$NaturalThing
0.8889	0.1111	0.8889	0.8889	0.9412	- #$NaturalThing

=== Confusion Matrix: #$NaturalThing 1 ===

+Class	-Class	<-----Classified as
TP: 0013	FP: 0000	+ #$NaturalThing
FN: 0002	TN: 0016	- #$NaturalThing

==========TEST ERROR==========
Correctly Classified Instances 2993.5483870967742 %
Incorrectly Classified Instances 26.45161290322581 %
Total Number of Instances 31

TestArticle:Ronald Reagan TargetConcept :#$NaturalThing classDecision = T confidence:27846.7213817851

```
Processing of #$CombustionInsturment
expand node: #$CombustionInstrument
actual_text_classifier( Ronald Reagan , #$CombustionInstrument )
classifyText( TargetConcept =#$CombustionInstrument , testArticle=Ronald Reagan)
---------------- Collect Sets -------------
Checking TopicFile: CombustionInstrument.tpc.txt
Computing from scratch: #$CombustionInstrument
Loading Positive Reference Examples
+ NIL
Loading Negative Reference Examples
-
- #$wikip-JapaneseLanguage
  |--> loaded Japanese_language ***
- #$wikip-Veganism
  |--> loaded Veganism ***
- #$wikip-Wicca
  |--> loaded Wicca ***
- #$wikip-Murder
  |--> loaded Murder ***
- #$wikip-SafetyEngineering
  |--> loaded Safety_engineering ***
Using Total Positive Sampling  pc = 33 pt=
+
+ #$wikip-Match
  |--> loaded Match ***
+ #$wikip-MagnifyingGlass
+ #$wikip-FuseExplosives
+ #$wikip-IncendiaryDevice
  |--> loaded Incendiary_device ***
+ #$wikip-Lighter
  |--> loaded Lighter ***
+ #$wikip-PilotLight
+ #$wikip-PropaneTorch
+ #$wikip-FlarePyrotechnic
+ #$wikip-Firework
  |--> loaded Firework ***
+ #$wikip-Pyrotechnics
  |--> loaded Pyrotechnics ***
+ #$wikip-Ethanol
  |--> loaded Ethanol ***
+ #$wikip-Methane
  |--> loaded Methane ***
+ #$wikip-Candle
  |--> loaded Candle ***
+ #$wikip-Methanol
```

|--> loaded Methanol ***
+ #$wikip-Propane
|--> loaded Propane ***
+ #$wikip-Napalm
|--> loaded Napalm ***
+ #$wikip-Pantiliner
+ #$wikip-Blueprint
|--> loaded Blueprint ***
+ #$wikip-FacialTissue
+ #$wikip-Poster
|--> loaded Poster ***
+ #$wikip-MoneyOrder
|--> loaded Money_order ***
+ #$wikip-Cheque
+ #$wikip-Category:WritingPaper
+ #$wikip-BusinessCard
|--> loaded Business_card ***
+ #$wikip-Paper
|--> loaded Paper ***
+ #$wikip-SanitaryNapkin
+ #$wikip-BountyPaperTowel
+ #$wikip-PaperTowel
|--> loaded Paper_towel ***
+ #$wikip-Tampon
|--> loaded Tampon ***
+ #$wikip-Cardboard
|--> loaded Cardboard ***
+ #$wikip-ToiletPaper
|--> loaded Toilet_paper ***
+ #$wikip-IndexCard
Using Random Negative Sampling
- #$wikip-Mapudungun
|--> loaded Mapudungun
- #$wikip-Prometheus
|--> loaded Prometheus
- #$wikip-LesothoLoti
- #$wikip-NicobareseLanguages
- #$wikip-ArtTherapy
- #$wikip-ZarmaLanguage
|--> loaded Zarma_language
- #$wikip-SicilianLanguage
|--> loaded Sicilian_language
- #$wikip-Hades
|--> loaded Hades
- #$wikip-Tool

|--> loaded Tool
- #$wikip-Pragmatics
|--> loaded Pragmatics
- #$wikip-SinNombreVirus
- #$wikip-GraphicDesign
|--> loaded Graphic_design
- #$wikip-Needlepoint
|--> loaded Needlepoint
- #$wikip-PhilosophyOfLogic
- #$wikip-Skyscraper
|--> loaded Skyscraper
- #$wikip-Lutheranism
|--> loaded Lutheranism
- #$wikip-LebanesePound
- #$wikip-Deconstruction
|--> loaded Deconstruction
- #$wikip-MiddlePersian
- #$wikip-Prolog
|--> loaded Prolog
- #$wikip-VoticLanguage
|--> loaded Votic_language
- #$wikip-OsseticLanguage
|--> loaded Ossetic_language
- #$wikip-Listeriosis
- #$wikip-Castle
|--> loaded Castle
- #$wikip-Relaxation
|--> loaded Relaxation
- #$wikip-Environmentalism
- #$wikip-Hades
|--> loaded Hades
- #$wikip-UdmurtLanguage
- #$wikip-Breakdance
|--> loaded Breakdance
- #$wikip-TrainStation
|--> loaded Train_station
- #$wikip-Vorlin
- #$wikip-YaghnobiLanguage
- #$wikip-Kilometer
- #$wikip-RheaMythology
- #$wikip-TarifitLanguage
- #$wikip-Circle
|--> loaded Circle
- #$wikip-RajasthaniLanguage
- #$wikip-ZuniLanguage

- #$wikip-ArchitecturalDesigner
- #$wikip-YavapaiLanguage
- #$wikip-BiblicalHebrewLanguage
- #$wikip-Nervousness
- #$wikip-Zoroastrianism
  |--> loaded Zoroastrianism
- #$wikip-SicilianLanguage
  |--> loaded Sicilian_language
- #$wikip-AkanLanguages
  |--> loaded Akan_languages
- #$wikip-Pentecostalism
  |--> loaded Pentecostalism
- #$wikip-Epistemology
  |--> loaded Epistemology
- #$wikip-ThaiBahtBaht
- #$wikip-BasicEnglish
  |--> loaded Basic_English
- #$wikip-Centime
  |--> loaded Centime
- #$wikip-YiLanguage
- #$wikip-MilesPerHour
  |--> loaded Miles_per_hour
- #$wikip-Calculus
  |--> loaded Calculus
Using Total Negative Sampling
---------- Build Models ----------
Second pass Verification ----------
Test +: Test +:#$wikip-PropaneTorch discriminate_article(article =Propane_torch)

ARTICLE:[Propane_torch]
ARTICLE URL: http://localhost/cyclopedia/index.php/Special:Export/Propane_torch
pick_best_lang_bigram_words(...)
P=-1.4041355177697 N=-0.281805060798581
ANALYSIS :Negative 1.12233045697112: Propane_torch
skipped
Test +:#$wikip-Firework discriminate_article(article =Firework)

ARTICLE:[Firework]
ARTICLE URL: http://localhost/cyclopedia/index.php/Special:Export/Firework
pick_best_lang_bigram_words(...)
P=-72704.1273499851 N=-102717.018734567
ANALYSIS :Positive 30012.8913845819: Firework
skipped
Test +:#$wikip-Pyrotechnics discriminate_article(article =Pyrotechnics)

ARTICLE:[Pyrotechnics]
ARTICLE URL: http://localhost/cyclopedia/index.php/Special:Export/Pyrotechnics
pick_best_lang_bigram_words(...)
P=-9028.82057427098 N=-12721.0726375391
ANALYSIS :Positive 3692.25206326809: Pyrotechnics
skipped
Test +:#$wikip-Ethanol discriminate_article(article =Ethanol)

ARTICLE:[Ethanol]
ARTICLE URL: http://localhost/cyclopedia/index.php/Special:Export/Ethanol
pick_best_lang_bigram_words(...)
P=-186875.199945972 N=-285509.013375938
ANALYSIS :Positive 98633.8134299663: Ethanol
skipped
Test +:#$wikip-Methane discriminate_article(article =Methane)

ARTICLE:[Methane]
ARTICLE URL: http://localhost/cyclopedia/index.php/Special:Export/Methane
pick_best_lang_bigram_words(...)
P=-91205.8551867806 N=-138332.931298671
ANALYSIS :Positive 47127.0761118903: Methane
skipped
Test +:#$wikip-Pantiliner discriminate_article(article =Pantiliner)

ARTICLE:[Pantiliner]
ARTICLE URL: http://localhost/cyclopedia/index.php/Special:Export/Pantiliner
pick_best_lang_bigram_words(...)
P=-1.4041355177697 N=-0.281805060798581
ANALYSIS :Negative 1.12233045697112: Pantiliner
skipped
Test +:#$wikip-Poster discriminate_article(article =Poster)

ARTICLE:[Poster]
ARTICLE URL: http://localhost/cyclopedia/index.php/Special:Export/Poster
pick_best_lang_bigram_words(...)
P=-32601.4189351849 N=-44909.0642123437
ANALYSIS :Positive 12307.6452771588: Poster
skipped
Test +:#$wikip-MoneyOrder discriminate_article(article =Money_order)

ARTICLE:[Money_order]
ARTICLE URL: http://localhost/cyclopedia/index.php/Special:Export/Money_order
pick_best_lang_bigram_words(...)
P=-42252.8634620741 N=-58593.565995481
ANALYSIS :Positive 16340.7025334069: Money_order

skipped
Test +:#$wikip-SanitaryNapkin discriminate_article(article =Sanitary_napkin)

ARTICLE:[Sanitary_napkin]
ARTICLE URL: http://localhost/cyclopedia/index.php/Special:Export/Sanitary_napkin
pick_best_lang_bigram_words(...)
P=-1.4041355177697 N=-0.281805060798581
ANALYSIS :Negative 1.12233045697112: Sanitary_napkin
skipped
Test +:#$wikip-ToiletPaper discriminate_article(article =Toilet_paper)

ARTICLE:[Toilet_paper]
ARTICLE URL: http://localhost/cyclopedia/index.php/Special:Export/Toilet_paper
pick_best_lang_bigram_words(...)
P=-87272.9792236049 N=-122586.658134329
ANALYSIS :Positive 35313.6789107237: Toilet_paper
skipped
Test -:#$wikip-Hunan discriminate_article(article =Hunan)

ARTICLE:[Hunan]
ARTICLE URL: http://localhost/cyclopedia/index.php/Special:Export/Hunan
pick_best_lang_bigram_words(...)
P=-70848.7665918185 N=-70102.5140016593
ANALYSIS :Negative 746.252590159245: Hunan
skipped
Test -:#$wikip-Soling discriminate_article(article =Soling)

ARTICLE:[Soling]
ARTICLE URL: http://localhost/cyclopedia/index.php/Special:Export/Soling
pick_best_lang_bigram_words(...)
P=-1.4041355177697 N=-0.281805060798581
ANALYSIS :Negative 1.12233045697112: Soling
skipped
Test -:#$wikip-BubonicPlague discriminate_article(article =Bubonic_plague)

ARTICLE:[Bubonic_plague]
ARTICLE URL: http://localhost/cyclopedia/index.php/Special:Export/Bubonic_plague

pick_best_lang_bigram_words(...)
P=-169743.760107839 N=-169843.309252151
ANALYSIS :Positive 99.5491443128558: Bubonic_plague
skipped
Test -:#$wikip-Karma discriminate_article(article =Karma)

ARTICLE:[Karma]

ARTICLE URL: http://localhost/cyclopedia/index.php/Special:Export/Karma
pick_best_lang_bigram_words(...)
P=-93288.9281807925 N=-91871.469584935
ANALYSIS   :Negative 1417.45859585745: Karma
 skipped
Test -:#$wikip-Environmentalism discriminate_article(article =Environmentalism)

ARTICLE:[Environmentalism]
ARTICLE URL: http://localhost/cyclopedia/index.php/Special:Export/Environmentalism
pick_best_lang_bigram_words(...)
P=-1.4041355177697 N=-0.281805060798581
ANALYSIS   :Negative 1.12233045697112: Environmentalism skipped
Test -:#$wikip-Dendrochronology discriminate_article(article =Dendrochronology)

ARTICLE:[Dendrochronology]
ARTICLE URL: http://localhost/cyclopedia/index.php/Special:Export/Dendrochronology
pick_best_lang_bigram_words(...)
P=-42750.3919136621 N=-42929.4873319069
ANALYSIS   :Positive 179.095418244848: Dendrochronology
  -------------> Adding -Example Dendrochronology ***
Test -:#$wikip-SwaziLilangeni discriminate_article(article =Swazi_lilangeni)

ARTICLE:[Swazi_lilangeni]
ARTICLE URL: http://localhost/cyclopedia/index.php/Special:Export/Swazi_lilangeni
pick_best_lang_bigram_words(...)
P=-10424.7067575251 N=-10202.4398657798
ANALYSIS   :Negative 222.266891745348: Swazi_lilangeni skipped
Test -:#$wikip-Ounce discriminate_article(article =Ounce)

ARTICLE:[Ounce]
ARTICLE URL: http://localhost/cyclopedia/index.php/Special:Export/Ounce
pick_best_lang_bigram_words(...)
P=-26063.8786615862 N=-26022.2859995001
ANALYSIS   :Negative 41.5926620860773: Ounce skipped
Test -:#$wikip-WebOntologyLanguage discriminate_article(article =Web_Ontology_Language)

ARTICLE:[Web_Ontology_Language]
ARTICLE URL: http://localhost/cyclopedia/index.php/Special:Export/Web_Ontology_Language
pick_best_lang_bigram_words(...)
P=-42152.637025844 N=-39103.1299650178
ANALYSIS   :Negative 3049.5070608262: Web_Ontology_Language skipped
Test -:#$wikip-Merman discriminate_article(article =Merman)

ARTICLE:[Merman]
ARTICLE URL: http://localhost/cyclopedia/index.php/Special:Export/Merman

pick_best_lang_bigram_words(...)
P=-14887.0720137541 N=-14093.922439144
ANALYSIS   :Negative 793.149574610175: Merman
 skipped
Test -:#$wikip-PawneeLanguage discriminate_article(article =Pawnee_language)

ARTICLE:[Pawnee_language]
ARTICLE URL: http://localhost/cyclopedia/index.php/Special:Export/Pawnee_languag
e
pick_best_lang_bigram_words(...)
P=-1.4041355177697 N=-0.281805060798581
ANALYSIS   :Negative 1.12233045697112: Pawnee_language
 skipped
Test -:#$wikip-BarrelUnit discriminate_article(article =Barrel_unit)

ARTICLE:[Barrel_unit]
ARTICLE URL: http://localhost/cyclopedia/index.php/Special:Export/Barrel_unit
pick_best_lang_bigram_words(...)
P=-1.4041355177697 N=-0.281805060798581
ANALYSIS   :Negative 1.12233045697112: Barrel_unit
 skipped
Test -:#$wikip-BambaraLanguage discriminate_article(article =Bambara_language)

ARTICLE:[Bambara_language]
ARTICLE URL: http://localhost/cyclopedia/index.php/Special:Export/Bambara_langua
ge
pick_best_lang_bigram_words(...)
P=-61840.4832298856 N=-57481.6372020606
ANALYSIS   :Negative 4358.84602782502: Bambara_language
 skipped
Test -:#$wikip-ThaiBahtBaht discriminate_article(article =Thai_baht_Baht)

ARTICLE:[Thai_baht_Baht]
ARTICLE URL: http://localhost/cyclopedia/index.php/Special:Export/Thai_baht_Baht

pick_best_lang_bigram_words(...)
P=-1.4041355177697 N=-0.281805060798581
ANALYSIS   :Negative 1.12233045697112: Thai_baht_Baht
 skipped
Test -:#$wikip-Enjoyment discriminate_article(article =Enjoyment)

ARTICLE:[Enjoyment]
ARTICLE URL: http://localhost/cyclopedia/index.php/Special:Export/Enjoyment
pick_best_lang_bigram_words(...)
P=-22.9847583224305 N=-22.9847583224305

ANALYSIS :Positive 0: Enjoyment skipped
Test -:#$wikip-Zaghawa discriminate_article(article =Zaghawa)

ARTICLE:[Zaghawa]
ARTICLE URL: http://localhost/cyclopedia/index.php/Special:Export/Zaghawa
pick_best_lang_bigram_words(...)
P=-1.4041355177697 N=-0.281805060798581
ANALYSIS :Negative 1.12233045697112: Zaghawa skipped
Test -:#$wikip-Paleontology discriminate_article(article =Paleontology)

ARTICLE:[Paleontology]
ARTICLE URL: http://localhost/cyclopedia/index.php/Special:Export/Paleontology
pick_best_lang_bigram_words(...)
P=-42166.7568964526 N=-41549.9702738539
ANALYSIS :Negative 616.786622598738: Paleontology skipped
Test -:#$wikip-Socialism discriminate_article(article =Socialism)

ARTICLE:[Socialism]
ARTICLE URL: http://localhost/cyclopedia/index.php/Special:Export/Socialism
pick_best_lang_bigram_words(...)
P=-181281.898871458 N=-178073.533051144
ANALYSIS :Negative 3208.36582031398: Socialism skipped
Test -:#$wikip-TurkmenistaniManat discriminate_article(article =Turkmenistani_manat)

ARTICLE:[Turkmenistani_manat]
ARTICLE URL: http://localhost/cyclopedia/index.php/Special:Export/Turkmenistani_manat
pick_best_lang_bigram_words(...)
P=-1.4041355177697 N=-0.281805060798581
ANALYSIS :Negative 1.12233045697112: Turkmenistani_manat skipped
Test -:#$wikip-Spouse discriminate_article(article =Spouse)

ARTICLE:[Spouse]
ARTICLE URL: http://localhost/cyclopedia/index.php/Special:Export/Spouse
pick_best_lang_bigram_words(...)
P=-22.9847583224305 N=-22.9847583224305
ANALYSIS :Positive 0: Spouse skipped
Test -:#$wikip-BlackfootLanguage discriminate_article(article =Blackfoot_language)

ARTICLE:[Blackfoot_language]
ARTICLE URL: http://localhost/cyclopedia/index.php/Special:Export/Blackfoot_language
pick_best_lang_bigram_words(...)
P=-42330.105287583 N=-43383.9052248139
ANALYSIS :Positive 1053.79993723094: Blackfoot_language
----------> Adding -Example Blackfoot_language ***
Test -:#$wikip-Second discriminate_article(article =Second)

ARTICLE:[Second]
ARTICLE URL: http://localhost/cyclopedia/index.php/Special:Export/Second
pick_best_lang_bigram_words(...)
P=-40583.2331338127 N=-40436.4713502932
ANALYSIS   :Negative 146.761783519498: Second skipped
Test -:#$wikip-BolivianPeso discriminate_article(article =Bolivian_peso)

ARTICLE:[Bolivian_peso]
ARTICLE URL: http://localhost/cyclopedia/index.php/Special:Export/Bolivian_peso
pick_best_lang_bigram_words(...)
P=-1.4041355177697 N=-0.281805060798581
ANALYSIS   :Negative 1.12233045697112: Bolivian_peso skipped
Quiz : #$CombustionInstrument 1
  QPS:1 QPR: 7 QPC:7
  QNS:0.846153846153846 QNR: 11 QNC:13
  QCS:0.9 QCR: 18 QCC:20
TargetConcept :#$CombustionInstrument
testListFile :
resultListFile :ArticleClassList.CombustionInstrument.txt
csvListFile :ArticleClassList.CombustionInstrument.csv
 TotalCount = PositiveCount + NegativeCount
 595858 = 144351 + 451507
====================== DIFFERENCE ========================
discriminate_article(article =Ronald Reagan)

ARTICLE:[Ronald Reagan]
ARTICLE URL: http://localhost/cyclopedia/index.php/Special:Export/Ronald Reagan
pick_best_lang_bigram_words(...)
P=-662036.453504916 N=-651015.189319758
ANALYSIS   :Negative 11021.2641851582: Ronald Reagan
   TargetConcept :#$CombustionInstrument classDecision = F
Profiling Classifier :#$CombustionInstrument
Flushing Tables to Disk
Untieing
Removing Backups
Compressing data to CombustionInstrument.tpr.rar

RAR 3.51   Copyright (c) 1993-2005 Alexander Roshal   7 Oct 2005
Shareware version   Type RAR -? for help

Evaluation copy. Please register.

Creating archive CombustionInstrument.tpr.rar

```
Adding    CombustionInstrument.tpc.nc                    OK
Adding    CombustionInstrument.tpc.ndb                    OK
Adding    CombustionInstrument.tpc.pc                    OK
Adding    CombustionInstrument.tpc.pdb                    OK
Adding    CombustionInstrument.tpc.txt                    OK
Deleting CombustionInstrument.tpc.txt      deleted
Deleting CombustionInstrument.tpc.pdb      deleted
Deleting CombustionInstrument.tpc.pc       deleted
Deleting CombustionInstrument.tpc.ndb      deleted
Deleting CombustionInstrument.tpc.nc       deleted
Done
Done #$CombustionInstrument
STOP: Ronald Reagan not in #$CombustionInstrument - -
```

APPENDIX B

KE-TEXT DEFINING DOMAIN

```
Default Mt: UniversalVocabularyMt.

Constant:wikiSalientURL.

isa :  BinaryPredicate .
genlPreds :  salientURI.
arity :  2  .
arg1Isa :  Thing  .
arg2Isa :  UniformResourceLocator  .


Constant: positiveWikiExample.
isa: BinaryPredicate.
arg1Isa: Thing.
arg2Isa: WikiArticle.

Constant: negativeWikiExample .
isa: BinaryPredicate.
arg1Isa: Thing.
arg2Isa: WikiArticle.


f:(implies
                                    (and            (topicOfPCW ?W ?C)
                                                    (isa ?W WikiArticle)
                                                    ( or        (isa ?C ?TargetConcept)
(genls ?C ?TargetConcept)))
(positiveWikiExample ?TargetConcept ?W)).

;; f:(implies
;;                                  (and            (topicOfPCW ?W ?C)
;;                                                  (isa ?W WikiArticle)
;;                                                  (disjointWith ?C ?TargetConcept))
;;(negativeWikiExample ?TargetConcept ?W)).

f:(implies
                                    (and
                                                    (genls ?TargetConcept ?Something)
                                                    (genls ?C ?something)
                                                    (not (genls ?c ?TargetConcept))
                                                    (topicOfPCW ?W ?C)
                                                    (isa ?W WikiArticle)
                                        )
(negativeWikiExample ?TargetConcept ?W)).
```

```
f:(implies
                    (and
                        (genls ?C Person)
                        (not (genls ?c ?TargetConcept))
                        (topicOfPCW ?W ?C)
                        (isa ?W WikiArticle)
                    )
(negativeWikiExample ?TargetConcept ?W)).

f:(implies
                    (and
                        (genls ?C EnduringThing-Localized)
                        (not (genls ?c ?TargetConcept))
                        (topicOfPCW ?W ?C)
                        (isa ?W WikiArticle)
                    )
(negativeWikiExample ?TargetConcept ?W)).

Constant: positiveExampleURL .
isa: BinaryPredicate.
arg1Isa: Thing.
arg2Isa: UniformResourceLocator.

Constant: negativeExampleURL.
isa: BinaryPredicate.
arg1Isa: Thing.
arg2Isa: UniformResourceLocator.



f:(implies
                    (and
                        ( or (isa ?C ?TargetConcept)(genls ?C ?

TargetConcept))
                        (topicOfPCW ?W ?C)
                        (isa ?W WikiArticle)
                        (salientURL ?W ?URL)
                )
(positiveExampleURL ?TargetConcept ?URL)).

f:(implies
                    (and
                        (disjointWith ?C ?TargetConcept)
```

```
                                        (topicOfPCW ?W ?C)
                                        (isa ?W WikiArticle)
                                        (salientURL ?W ?URL)
)
(negativeExampleURL ?TargetConcept ?URL)).

Constant: positiveExampleTitle .
isa: BinaryPredicate.
arg1Isa: Thing.
arg2Isa: UniformResourceLocator.
f:(implies
                        (and
                                        ( or        (isa ?C ?TargetConcept)
(genls ?C ?TargetConcept))
                                        (topicOfPCW ?W ?C)
                                        (isa ?W WikiArticle)
                                        (titleOfWork ?W ?TITLE )
)
(positiveExampleTitle  ?TargetConcept ?TITLE )).



Constant: positivePCWExample.
isa: BinaryPredicate.
arg1Isa: Thing.
arg2Isa: Thing.

Constant: negativePCWExample .
isa: BinaryPredicate.
arg1Isa: Thing.
arg2Isa: Thing.


f:(implies
                        (and        (topicOfPCW ?W ?C)
                                    ( or (isa ?C ?TargetConcept)
    (genls ?C ?TargetConcept)))
(positivePCWExample ?TargetConcept ?W)).

f:(implies
                        (and        (topicOfPCW ?W ?C)
                                    (not ( or (isa ?C ?TargetConcept)
                    (genls ?C ?TargetConcept))))
(negativePCWExample ?TargetConcept ?W)).
```

```
Constant: positivePCWReferenceExample.
isa: BinaryPredicate.
arg1Isa: Thing.
arg2Isa: Thing.

Constant: negativePCWReferenceExample .
isa: BinaryPredicate.
arg1Isa: Thing.
arg2Isa: Thing.

Constant: ReferenceExample .
isa: Collection.
genls: WikiArticle.


f:(implies
                    (and
                              (isa ?W ReferenceExample )
                              (topicOfPCW ?W ?C)
                              ( or (isa ?C ?TargetConcept)(genls ?C ?

TargetConcept)))
(positivePCWReferenceExample ?TargetConcept ?W)).

f:(implies
                    (and
                              (isa ?W ReferenceExample )
                              (topicOfPCW ?W ?C)
                              (not ( or (isa ?C ?TargetConcept)(genls ?C ?

TargetConcept))))
(negativePCWReferenceExample ?TargetConcept ?W)).
```

```
;;-----------------------------------
;; Reference Examples
;;-----------------------------------
Constant:wikip-Psilocybin.
isa:ReferenceExample.
Constant:wikip-Tentacle.
isa:ReferenceExample.
Constant:wikip-HosniMubarak.
isa:ReferenceExample.
Constant:wikip-JapaneseLanguage.
isa:ReferenceExample.
Constant:wikip-Veganism.
isa:ReferenceExample.
Constant:wikip-MichaelCrichton.
isa:ReferenceExample.
Constant:wikip-Casablanca.
isa:ReferenceExample.
Constant:wikip-Volvo960.
isa:ReferenceExample.
Constant:wikip-Codine.
isa:ReferenceExample.
Constant:wikip-SanFranciscoCalifornia.
isa:ReferenceExample.
Constant:wikip-SubwayTrain.
isa:ReferenceExample.
Constant:wikip-Wicca.
isa:ReferenceExample.
Constant:wikip-Doughnut.
isa:ReferenceExample.
Constant:wikip-Murder.
isa:ReferenceExample.
Constant:wikip-Emu.
isa:ReferenceExample.
Constant:wikip-DukeEllington.
isa:ReferenceExample.
Constant:wikip-SafetyEngineering.
isa:ReferenceExample.
Constant:wikip-PlasticBag.
isa:ReferenceExample.
Constant:wikip-Cerebellum.
isa:ReferenceExample.
Constant:wikip-Salamander.
isa:ReferenceExample.
Constant:wikip-WaterBuffalo.
isa:ReferenceExample.
;;-----------------------------------
;; Classes that can have a TextClassifier
;; defined for them
;;-----------------------------------
Constant: TextClassifierClass .
isa: Collection.

Constant:Dog.
isa: TextClassifierClass .
Constant:Cat.
isa: TextClassifierClass .
Constant:Animal.
isa: TextClassifierClass .
Constant:MaleHuman.
isa: TextClassifierClass .
Constant:FemaleHuman.
isa: TextClassifierClass .
Constant:Organism-Whole.
isa: TextClassifierClass .
Constant:Mammal.
isa: TextClassifierClass .
Constant:Food.
isa: TextClassifierClass .
Constant:Plant.
isa: TextClassifierClass .
Constant:Alive.
isa: TextClassifierClass .
Constant:AquaticOrganism.
isa: TextClassifierClass .
Constant:Author.
isa: TextClassifierClass .
Constant:Automobile.
isa: TextClassifierClass .
Constant:HobbyActivity.
isa: TextClassifierClass .
Constant:Aristocrat.
isa: TextClassifierClass .
Constant:Professional.
isa: TextClassifierClass .
Constant:Politician.
isa: TextClassifierClass .
Constant:ProductType.
isa: TextClassifierClass .
```

```
Constant:RegionTypeByTerrain.
isa: TextClassifierClass .
Constant:Leader.
isa: TextClassifierClass .
Constant:FamousIndividual.
isa: TextClassifierClass .
Constant:Enterainer.
isa: TextClassifierClass .
Constant:Artist.
isa: TextClassifierClass .
Constant:UnitedStatesPerson.
isa: TextClassifierClass .
Constant:Artifact-HumanCreated.
isa: TextClassifierClass .
Constant:Container.
isa: TextClassifierClass .
Constant:CuturalThing.
isa: TextClassifierClass .
Constant:FulePoweedDevice.
isa: TextClassifierClass .
Constant:Individual.
isa: TextClassifierClass .
Constant:Collection.
isa: TextClassifierClass .
Constant:GeopoliticalEntity.
isa: TextClassifierClass .
Constant:Place.
isa: TextClassifierClass .
Constant:Religon.
isa: TextClassifierClass .
Constant:BeliefSystem.
isa: TextClassifierClass .
Constant:ChemicalSubstanceType.
isa: TextClassifierClass .
Constant:NaturalThing.
isa: TextClassifierClass .
Constant:BiochemicallyHarmfulSubstance.
isa: TextClassifierClass .
Constant:MedicalProcedure.
isa: TextClassifierClass .
Constant:WeatherEvent.
isa: TextClassifierClass .
Constant:AstronomicalObject.
isa: TextClassifierClass .
```

```
;;---------------------------------
;; from a random sample census of wikip
;;---------------------------------
;; #$Agent-Generic => 1739
 Constant: #$Agent-Generic .
   isa: TextClassifierClass .

;; #$Collection => 1665
 Constant: #$Collection .
   isa: TextClassifierClass .

;; #$Individual => 1151
 Constant: #$Individual .
   isa: TextClassifierClass .

;; #$Artifact-Generic => 971
 Constant: #$Artifact-Generic .
   isa: TextClassifierClass .

;; #$Agent-Underspecified => 951
 Constant: #$Agent-Underspecified .
   isa: TextClassifierClass .

;; #$BiologicalLivingObject => 884
 Constant: #$BiologicalLivingObject .
   isa: TextClassifierClass .

;; #$Landmark-Underspecified => 850
;; Constant: #$Landmark-Underspecified .
;;   isa: TextClassifierClass .

;; #$InformationStore => 760
 Constant: #$InformationStore .
   isa: TextClassifierClass .

;; #$HumanScaleObject => 753
 Constant: #$HumanScaleObject .
   isa: TextClassifierClass .

;; #$IntelligentAgent => 662
 Constant: #$IntelligentAgent .
   isa: TextClassifierClass .

;; #$Artifact => 628
 Constant: #$Artifact .
```

```
  isa: TextClassifierClass .

;; #$Animal => 625
 Constant: #$Animal .
  isa: TextClassifierClass .

;; #$PropositionalConceptualWork => 495
 Constant: #$PropositionalConceptualWork .
  isa: TextClassifierClass .

;; #$CulturalThing => 486
 Constant: #$CulturalThing .
  isa: TextClassifierClass .

;; #$NonNaturalThing => 478
 Constant: #$NonNaturalThing .
  isa: TextClassifierClass .

;; #$Person => 460
 Constant: #$Person .
  isa: TextClassifierClass .

;; #$GeographicalRegion => 458
 Constant: #$GeographicalRegion .
  isa: TextClassifierClass .

;; #$InanimateObject => 450
 Constant: #$InanimateObject .
  isa: TextClassifierClass .

;; #$City => 385
 Constant: #$City .
  isa: TextClassifierClass .

;; #$InformationBearingThing => 370
 Constant: #$InformationBearingThing .
  isa: TextClassifierClass .

;; #$HomoSapiens => 310
 Constant: #$HomoSapiens .
  isa: TextClassifierClass .

;; #$NaturalThing => 306
 Constant: #$NaturalThing .
  isa: TextClassifierClass .

;; #$DurableGood => 305
 Constant: #$DurableGood .
  isa: TextClassifierClass .

;; #$Artifact-NonAgentive => 302
 Constant: #$Artifact-NonAgentive .
  isa: TextClassifierClass .

;; #$PhysicalDevice => 276
 Constant: #$PhysicalDevice .
  isa: TextClassifierClass .

;; #$ObjectWithUse => 273
 Constant: #$ObjectWithUse .
  isa: TextClassifierClass .

;; #$SolidTangibleArtifact => 269
 Constant: #$SolidTangibleArtifact .
  isa: TextClassifierClass .

;; #$BodyOfWater => 268
 Constant: #$BodyOfWater .
  isa: TextClassifierClass .

;; #$NaturalTangibleStuff => 268
 Constant: #$NaturalTangibleStuff .
  isa: TextClassifierClass .

;; #$LandTopographicalFeature => 266
 Constant: #$LandTopographicalFeature .
  isa: TextClassifierClass .

;; #$Place => 254
 Constant: #$Place .
  isa: TextClassifierClass .

;; #$SocialBeing => 253
 Constant: #$SocialBeing .
  isa: TextClassifierClass .

;; #$LegalAgent => 253
 Constant: #$LegalAgent .
  isa: TextClassifierClass .
```

```
;; #$OrganicMaterial => 248
 Constant: #$OrganicMaterial .
  isa: TextClassifierClass .

;; #$GeographicalThing => 247
 Constant: #$GeographicalThing .
  isa: TextClassifierClass .

;; #$County => 229
 Constant: #$County .
  isa: TextClassifierClass .

;; #$Roadway => 229
 Constant: #$Roadway .
  isa: TextClassifierClass .

;; #$Province => 229
 Constant: #$Province .
  isa: TextClassifierClass .

;; #$Airport-Physical => 229
 Constant: #$Airport-Physical .
  isa: TextClassifierClass .

;; #$Continent => 229
 Constant: #$Continent .
  isa: TextClassifierClass .

;; #$FictionalCharacter => 221
 Constant: #$FictionalCharacter .
  isa: TextClassifierClass .

;; #$AnimalBLO => 220
 Constant: #$AnimalBLO .
  isa: TextClassifierClass .

;; #$Artifact-HumanCreated => 218
 Constant: #$Artifact-HumanCreated .
  isa: TextClassifierClass .

;; #$ManMadeThing => 218
 Constant: #$ManMadeThing .
  isa: TextClassifierClass .

;; #$ArtifactType => 186
 Constant: #$ArtifactType .
  isa: TextClassifierClass .

;; #$ManufacturedGoods => 182
 Constant: #$ManufacturedGoods .
  isa: TextClassifierClass .

;; #$Product => 179
 Constant: #$Product .
  isa: TextClassifierClass .

;; #$RoadVehicle => 173
 Constant: #$RoadVehicle .
  isa: TextClassifierClass .

;; #$Organization => 169
 Constant: #$Organization .
  isa: TextClassifierClass .

;; #$Technology-Artifact => 160
 Constant: #$Technology-Artifact .
  isa: TextClassifierClass .

;; #$Alive => 156
 Constant: #$Alive .
  isa: TextClassifierClass .

;; #$EukaryoticOrganism => 156
 Constant: #$EukaryoticOrganism .
  isa: TextClassifierClass .

;; #$PerceptualAgent => 156
 Constant: #$PerceptualAgent .
  isa: TextClassifierClass .

;; #$GeographicalAgent => 156
 Constant: #$GeographicalAgent .
  isa: TextClassifierClass .

;; #$Sentient => 156
 Constant: #$Sentient .
  isa: TextClassifierClass .

;; #$Agent-NonGeographical => 156
 Constant: #$Agent-NonGeographical .
```

```
  isa: TextClassifierClass .

;; #$PerceptualAgent-Embodied => 156
 Constant: #$PerceptualAgent-Embodied .
  isa: TextClassifierClass .

;; #$Artifact-Agentive => 156
 Constant: #$Artifact-Agentive .
  isa: TextClassifierClass .

;; #$Organism-Whole => 156
 Constant: #$Organism-Whole .
  isa: TextClassifierClass .

;; #$GeopoliticalEntity => 156
 Constant: #$GeopoliticalEntity .
  isa: TextClassifierClass .

;; #$SentientAnimal => 156
 Constant: #$SentientAnimal .
  isa: TextClassifierClass .

;; #$Vertebrate => 156
 Constant: #$Vertebrate .
  isa: TextClassifierClass .

;; #$Municipality => 156
 Constant: #$Municipality .
  isa: TextClassifierClass .

;; #$AirBreathingVertebrate => 147
 Constant: #$AirBreathingVertebrate .
  isa: TextClassifierClass .

;; #$PoweredDevice => 142
 Constant: #$PoweredDevice .
  isa: TextClassifierClass .

;; #$TerrestrialOrganism => 134
 Constant: #$TerrestrialOrganism .
  isa: TextClassifierClass .

;; #$Train-TransportationDevice => 116
 Constant: #$Train-TransportationDevice .
  isa: TextClassifierClass .

;; #$MaleHuman => 127
 Constant: #$MaleHuman .
  isa: TextClassifierClass .

;; #$ChemicalSubstanceType => 114
 Constant: #$ChemicalSubstanceType .
  isa: TextClassifierClass .

;; #$Container => 110
 Constant: #$Container .
  isa: TextClassifierClass .

;; #$Omnivore => 98
 Constant: #$Omnivore .
  isa: TextClassifierClass .

;; #$PersonWithOccupation => 84
 Constant: #$PersonWithOccupation .
  isa: TextClassifierClass .

;; #$FemaleHuman => 83
 Constant: #$FemaleHuman .
  isa: TextClassifierClass .

;; #$Conveyance => 81
 Constant: #$Conveyance .
  isa: TextClassifierClass .

;; #$TransportationDevice => 80
 Constant: #$TransportationDevice .
  isa: TextClassifierClass .

;; #$SelfPoweredDevice => 74
 Constant: #$SelfPoweredDevice .
  isa: TextClassifierClass .

;; #$TopographicalFeature => 72
 Constant: #$TopographicalFeature .
  isa: TextClassifierClass .

;; #$Octopus => 70
 Constant: #$Octopus .
  isa: TextClassifierClass .
```

```
;; #$PortableObject => 67
 Constant: #$PortableObject .
  isa: TextClassifierClass .

;; #$TransportationDevice-Vehicle => 65
 Constant: #$TransportationDevice-Vehicle .
  isa: TextClassifierClass .

;; #$OrganismPart => 65
 Constant: #$OrganismPart .
  isa: TextClassifierClass .

;; #$AnimalBodyRegion => 64
 Constant: #$AnimalBodyRegion .
  isa: TextClassifierClass .

;; #$TransportFacility => 64
 Constant: #$TransportFacility .
  isa: TextClassifierClass .

;; #$AnimalBodyPart => 64
 Constant: #$AnimalBodyPart .
  isa: TextClassifierClass .

;; #$ProductType => 62
 Constant: #$ProductType .
  isa: TextClassifierClass .

;; #$OrganismPartType => 62
 Constant: #$OrganismPartType .
  isa: TextClassifierClass .

;; #$Language => 60
 Constant: #$Language .
  isa: TextClassifierClass .

;; #$WheeledTransportationDevice => 60
 Constant: #$WheeledTransportationDevice .
  isa: TextClassifierClass .

;; #$HumanLanguage => 60
 Constant: #$HumanLanguage .
  isa: TextClassifierClass .

;; #$Automobile => 60
 Constant: #$Automobile .
  isa: TextClassifierClass .

;; #$LandTransportationVehicle => 60
 Constant: #$LandTransportationVehicle .
  isa: TextClassifierClass .

;; #$NonPersonAnimal => 59
 Constant: #$NonPersonAnimal .
  isa: TextClassifierClass .

;; #$NonHumanAnimal => 59
 Constant: #$NonHumanAnimal .
  isa: TextClassifierClass .

;; #$Device-SingleUser => 59
 Constant: #$Device-SingleUser .
  isa: TextClassifierClass .

;; #$Politician => 58
 Constant: #$Politician .
  isa: TextClassifierClass .

;; #$MaleAnimal => 57
 Constant: #$MaleAnimal .
  isa: TextClassifierClass .

;; #$IonTypeByChemicalSpecies => 57
 Constant: #$IonTypeByChemicalSpecies .
  isa: TextClassifierClass .

;; #$FuelPoweredDevice => 57
 Constant: #$FuelPoweredDevice .
  isa: TextClassifierClass .

;; #$EntertainmentOrArtsProfessional => 54
 Constant: #$EntertainmentOrArtsProfessional .
  isa: TextClassifierClass .

;; #$NonPoweredDevice => 53
 Constant: #$NonPoweredDevice .
  isa: TextClassifierClass .

;; #$NaturalLanguage => 53
```

```
Constant: #$NaturalLanguage .
  isa: TextClassifierClass .

;; #$InternalCombustionPoweredDevice => 53
Constant: #$InternalCombustionPoweredDevice .
  isa: TextClassifierClass .

;; #$ManufacturedGoodsType => 49
Constant: #$ManufacturedGoodsType .
  isa: TextClassifierClass .

;; #$AdultAnimal => 48
Constant: #$AdultAnimal .
  isa: TextClassifierClass .

;; #$RoadVehicle-GasolineEngine => 48
Constant: #$RoadVehicle-GasolineEngine .
  isa: TextClassifierClass .

;; #$SexuallyMature => 48
Constant: #$SexuallyMature .
  isa: TextClassifierClass .

;; #$RoadVehicle-InternalCombustionEngine => 48
Constant: #$RoadVehicle-InternalCombustionEngine .
  isa: TextClassifierClass .

;; #$Automobile-GasolineEngine => 48
Constant: #$Automobile-GasolineEngine .
  isa: TextClassifierClass .

;; #$InternalAnatomicalPart => 45
Constant: #$InternalAnatomicalPart .
  isa: TextClassifierClass .

;; #$InternalAnimalBodyRegionType => 45
Constant: #$InternalAnimalBodyRegionType .
  isa: TextClassifierClass .

;; #$MilitaryEquipment => 44
Constant: #$MilitaryEquipment .
  isa: TextClassifierClass .

;; #$ChemicalSpeciesType => 42
Constant: #$ChemicalSpeciesType .
  isa: TextClassifierClass .

;; #$TimeInterval => 42
Constant: #$TimeInterval .
  isa: TextClassifierClass .

;; #$PersonalProduct => 41
Constant: #$PersonalProduct .
  isa: TextClassifierClass .

;; #$HumanAdult => 41
Constant: #$HumanAdult .
  isa: TextClassifierClass .

;; #$SomethingToWear => 41
Constant: #$SomethingToWear .
  isa: TextClassifierClass .

;; #$LivingLanguage => 41
Constant: #$LivingLanguage .
  isa: TextClassifierClass .

;; #$FamousIndividual => 39
Constant: #$FamousIndividual .
  isa: TextClassifierClass .

;; #$FamousHuman => 39
Constant: #$FamousHuman .
  isa: TextClassifierClass .

;; #$FieldOfStudy => 37
Constant: #$FieldOfStudy .
  isa: TextClassifierClass .

;; #$SkilledPerson => 34
Constant: #$SkilledPerson .
  isa: TextClassifierClass .

;; #$CapitalCityOfRegion => 34
Constant: #$CapitalCityOfRegion .
  isa: TextClassifierClass .

;; #$DangerousThing => 34
Constant: #$DangerousThing .
  isa: TextClassifierClass .
```

```
;; #$Professional => 34
 Constant: #$Professional .
  isa: TextClassifierClass .

;; #$SkilledWorker => 34
 Constant: #$SkilledWorker .
  isa: TextClassifierClass .

;; #$DangerousTangibleThing => 33
 Constant: #$DangerousTangibleThing .
  isa: TextClassifierClass .

;; #$ElectricalDevice => 32
 Constant: #$ElectricalDevice .
  isa: TextClassifierClass .

;; #$InorganicMaterial => 32
 Constant: #$InorganicMaterial .
  isa: TextClassifierClass .

;; #$GovernmentEmployee => 29
 Constant: #$GovernmentEmployee .
  isa: TextClassifierClass .

;; #$PersonTypeByActivity => 28
 Constant: #$PersonTypeByActivity .
  isa: TextClassifierClass .

;; #$PersonTypeByOccupation => 28
 Constant: #$PersonTypeByOccupation .
  isa: TextClassifierClass .

;; #$MusicalPerformer => 28
 Constant: #$MusicalPerformer .
  isa: TextClassifierClass .

;; #$EdibleStuff => 28
 Constant: #$EdibleStuff .
  isa: TextClassifierClass .

;; #$NonAmericanCar => 28
 Constant: #$NonAmericanCar .
  isa: TextClassifierClass .

;; #$Entertainer => 28
 Constant: #$Entertainer .
  isa: TextClassifierClass .

;; #$DogTypeByBreed => 28
 Constant: #$DogTypeByBreed .
  isa: TextClassifierClass .

;; #$MilitaryWeapon => 27
 Constant: #$MilitaryWeapon .
  isa: TextClassifierClass .

;; #$Island => 27
 Constant: #$Island .
  isa: TextClassifierClass .

;; #$FoodOrDrink => 27
 Constant: #$FoodOrDrink .
  isa: TextClassifierClass .

;; #$FoodDrinkAndIngredients => 27
 Constant: #$FoodDrinkAndIngredients .
  isa: TextClassifierClass .

;; #$LandBody => 27
 Constant: #$LandBody .
  isa: TextClassifierClass .

;; #$LandMass => 27
 Constant: #$LandMass .
  isa: TextClassifierClass .

;; #$PublicOfficial => 26
 Constant: #$PublicOfficial .
  isa: TextClassifierClass .

;; #$Clothing-Generic => 25
 Constant: #$Clothing-Generic .
  isa: TextClassifierClass .

;; #$HeadOfState => 25
 Constant: #$HeadOfState .
  isa: TextClassifierClass .

;; #$ClothingItem => 25
```

```
Constant: #$ClothingItem .
  isa: TextClassifierClass .

;; #$CommunicationDevice => 23
Constant: #$CommunicationDevice .
  isa: TextClassifierClass .

;; #$River => 23
Constant: #$River .
  isa: TextClassifierClass .

;; #$ElectronicDevice => 23
Constant: #$ElectronicDevice .
  isa: TextClassifierClass .

;; #$Stream => 23
Constant: #$Stream .
  isa: TextClassifierClass .

;; #$NaturalObstacle => 23
Constant: #$NaturalObstacle .
  isa: TextClassifierClass .

;; #$WeaponType => 23
Constant: #$WeaponType .
  isa: TextClassifierClass .

;; #$Athlete => 22
Constant: #$Athlete .
  isa: TextClassifierClass .

Constant: #$WeaponSystem .
  isa: TextClassifierClass .

;; #$WesternHemispherePerson => 22
Constant: #$WesternHemispherePerson .
  isa: TextClassifierClass .

;; #$GovernmentRelatedEntity => 22
Constant: #$GovernmentRelatedEntity .
  isa: TextClassifierClass .

;; #$MilitaryHardware => 22
Constant: #$MilitaryHardware .
  isa: TextClassifierClass .

;; #$Weapon => 21
Constant: #$Weapon .
  isa: TextClassifierClass .

;; #$USCityOrCounty => 20
Constant: #$USCityOrCounty .
  isa: TextClassifierClass .

;; #$USCity => 20
Constant: #$USCity .
  isa: TextClassifierClass .

;; #$AmericanAutomobile => 20
Constant: #$AmericanAutomobile .
  isa: TextClassifierClass .

;; #$Artist => 18
Constant: #$Artist .
  isa: TextClassifierClass .

;; #$PoliticalParty => 18
Constant: #$PoliticalParty .
  isa: TextClassifierClass .

;; #$Carnivore => 18
Constant: #$Carnivore .
  isa: TextClassifierClass .

;; #$DomesticatedAnimal => 18
Constant: #$DomesticatedAnimal .
  isa: TextClassifierClass .

;; #$ExternalAnatomicalPart => 18
Constant: #$ExternalAnatomicalPart .
  isa: TextClassifierClass .

;; #$LegalGovernmentOrganization => 18
Constant: #$LegalGovernmentOrganization .
  isa: TextClassifierClass .

;; #$PortCity => 18
Constant: #$PortCity .
  isa: TextClassifierClass .
```

```
;; #$Food => 18
 Constant: #$Food .
  isa: TextClassifierClass .

;; #$TameAnimal => 18
 Constant: #$TameAnimal .
  isa: TextClassifierClass .

;; #$Employee => 17
 Constant: #$Employee .
  isa: TextClassifierClass .

;; #$PublicSectorEmployee => 17
 Constant: #$PublicSectorEmployee .
  isa: TextClassifierClass .

;; #$Leader => 17
 Constant: #$Leader .
  isa: TextClassifierClass .

;; #$Collectible => 17
 Constant: #$Collectible .
  isa: TextClassifierClass .

;; #$ConstructionArtifact => 17
 Constant: #$ConstructionArtifact .
  isa: TextClassifierClass .

;; #$HeadOfGovernment => 17
 Constant: #$HeadOfGovernment .
  isa: TextClassifierClass .

;; #$PharmaceuticalType => 17
 Constant: #$PharmaceuticalType .
  isa: TextClassifierClass .

;; #$DrugSubstance => 17
 Constant: #$DrugSubstance .
  isa: TextClassifierClass .

;; #$PersonWithNationality => 16
 Constant: #$PersonWithNationality .
  isa: TextClassifierClass .

;; #$EuropeanCar => 16
 Constant: #$EuropeanCar .
  isa: TextClassifierClass .

;; #$Author => 15
 Constant: #$Author .
  isa: TextClassifierClass .

;; #$Celebrity-Political => 15
 Constant: #$Celebrity-Political .
  isa: TextClassifierClass .

;; #$BiologicalSpecies => 15
 Constant: #$BiologicalSpecies .
  isa: TextClassifierClass .

;; #$Celebrity => 15
 Constant: #$Celebrity .
  isa: TextClassifierClass .

;; #$LuxuryItem => 15
 Constant: #$LuxuryItem .
  isa: TextClassifierClass .

;; #$BeliefSystem => 15
 Constant: #$BeliefSystem .
  isa: TextClassifierClass .

;; #$Writer => 15
 Constant: #$Writer .
  isa: TextClassifierClass .

;; #$Indo-EuropeanLanguageFamily => 15
 Constant: #$Indo-EuropeanLanguageFamily .
  isa: TextClassifierClass .

;; #$CanineAnimal => 14
 Constant: #$CanineAnimal .
  isa: TextClassifierClass .

;; #$Artist-Performer => 14
 Constant: #$Artist-Performer .
  isa: TextClassifierClass .

;; #$StateCapital => 14
```

```
 Constant: #$StateCapital .
   isa: TextClassifierClass .

;; #$Dog => 14
 Constant: #$Dog .
   isa: TextClassifierClass .

;; #$CanisGenus => 14
 Constant: #$CanisGenus .
   isa: TextClassifierClass .

;; #$AstronomicalObject => 14
 Constant: #$AstronomicalObject .
   isa: TextClassifierClass .

;; #$FemaleAnimal => 13
 Constant: #$FemaleAnimal .
   isa: TextClassifierClass .

;; #$BusinessPerson => 13
 Constant: #$BusinessPerson .
   isa: TextClassifierClass .

;; #$Reptile => 12
 Constant: #$Reptile .
   isa: TextClassifierClass .

;; #$Organ => 12
 Constant: #$Organ .
   isa: TextClassifierClass .

;; #$BodyOfWater-Large => 12
 Constant: #$BodyOfWater-Large .
   isa: TextClassifierClass .

;; #$ScaledAnimal => 12
 Constant: #$ScaledAnimal .
   isa: TextClassifierClass .

;; #$Electrolyte => 12
 Constant: #$Electrolyte .
   isa: TextClassifierClass .

;; #$Mineral => 12
 Constant: #$Mineral .
   isa: TextClassifierClass .

;; #$HumanOccupationConstruct => 11
 Constant: #$HumanOccupationConstruct .
   isa: TextClassifierClass .

;; #$HistoricHuman => 11
 Constant: #$HistoricHuman .
   isa: TextClassifierClass .

;; #$Tool => 11
 Constant: #$Tool .
   isa: TextClassifierClass .

;; #$ComputationalSystem => 11
 Constant: #$ComputationalSystem .
   isa: TextClassifierClass .

;; #$ProjectileWeaponOrLauncher => 11
 Constant: #$ProjectileWeaponOrLauncher .
   isa: TextClassifierClass .

;; #$MusclePoweredDevice => 11
 Constant: #$MusclePoweredDevice .
   isa: TextClassifierClass .

;; #$ComputerNetwork => 11
 Constant: #$ComputerNetwork .
   isa: TextClassifierClass .

;; #$Bird => 11
 Constant: #$Bird .
   isa: TextClassifierClass .

;; #$SportsOrganization => 11
 Constant: #$SportsOrganization .
   isa: TextClassifierClass .

;; #$InternalOrgan => 11
 Constant: #$InternalOrgan .
   isa: TextClassifierClass .

;; #$Workplace => 11
 Constant: #$Workplace .
   isa: TextClassifierClass .
```

```
;; #$ArtificialMaterial => 11
 Constant: #$ArtificialMaterial .
  isa: TextClassifierClass .

;; #$Metal => 11
 Constant: #$Metal .
  isa: TextClassifierClass .

;; #$MusicPerformanceAgent => 10
 Constant: #$MusicPerformanceAgent .
  isa: TextClassifierClass .

;; #$HerdAnimal => 10
 Constant: #$HerdAnimal .
  isa: TextClassifierClass .

;; #$Sea => 10
 Constant: #$Sea .
  isa: TextClassifierClass .

;; #$OuterGarment => 10
 Constant: #$OuterGarment .
  isa: TextClassifierClass .

;; #$Band-MusicGroup => 10
 Constant: #$Band-MusicGroup .
  isa: TextClassifierClass .

;; #$Device-UserPowered => 10
 Constant: #$Device-UserPowered .
  isa: TextClassifierClass .

;; #$Device-UserPowered => 10
 Constant: #$Device-UserPowered .
  isa: TextClassifierClass .

;; #$CombustionInstrument => 10
 Constant: #$CombustionInstrument .
  isa: TextClassifierClass .

;; #$Device-OneTimeUse => 10
 Constant: #$Device-OneTimeUse .
  isa: TextClassifierClass .

;; #$JapaneseCar => 10
 Constant: #$JapaneseCar .
  isa: TextClassifierClass .

;; #$Herbivore => 10
 Constant: #$Herbivore .
  isa: TextClassifierClass .

;; #$DeviceWithNoMovingParts => 10
 Constant: #$DeviceWithNoMovingParts .
  isa: TextClassifierClass .

;; #$ComputerHardwareComponent => 10
 Constant: #$ComputerHardwareComponent .
  isa: TextClassifierClass .

;; #$HandTool => 10
 Constant: #$HandTool .
  isa: TextClassifierClass .

;; #$ScienceAndNature-Topic => 10
 Constant: #$ScienceAndNature-Topic .
  isa: TextClassifierClass .

;; #$WildAnimal => 9
 Constant: #$WildAnimal .
  isa: TextClassifierClass .

;; #$UpperClass => 9
 Constant: #$UpperClass .
  isa: TextClassifierClass .

;; #$EuropeanCitizenOrSubject => 9
 Constant: #$EuropeanCitizenOrSubject .
  isa: TextClassifierClass .

;; #$DefenseSystem => 9
 Constant: #$DefenseSystem .
  isa: TextClassifierClass .

;; #$CelestialObject => 9
 Constant: #$CelestialObject .
  isa: TextClassifierClass .

;; #$BiochemicallyHarmfulSubstance => 9
```

```
Constant: #$BiochemicallyHarmfulSubstance .
  isa: TextClassifierClass .

;; #$Drink => 9
Constant: #$Drink .
  isa: TextClassifierClass .

;; #$Resources => 9
Constant: #$Resources .
  isa: TextClassifierClass .

;; #$ConsumableProduct => 9
Constant: #$ConsumableProduct .
  isa: TextClassifierClass .

;; #$CombustibleMaterial => 9
Constant: #$CombustibleMaterial .
  isa: TextClassifierClass .

;; #$Constellation => 9
Constant: #$Constellation .
  isa: TextClassifierClass .

;; #$UnalloyedMetal => 9
Constant: #$UnalloyedMetal .
  isa: TextClassifierClass .

;; #$LuxuryCar => 9
Constant: #$LuxuryCar .
  isa: TextClassifierClass .

;; #$SoccerPlayer => 9
Constant: #$SoccerPlayer .
  isa: TextClassifierClass .

;; #$MilitaryWeaponType => 9
Constant: #$MilitaryWeaponType .
  isa: TextClassifierClass .

;; #$CulturalActivity => 8
Constant: #$CulturalActivity .
  isa: TextClassifierClass .

;; #$TextileArtifact => 8
Constant: #$TextileArtifact .
  isa: TextClassifierClass .

;; #$GMAutomobile => 8
Constant: #$GMAutomobile .
  isa: TextClassifierClass .

;; #$ChryslerCar => 9
Constant: #$ChryslerCar .
  isa: TextClassifierClass .

;; #$Mountain => 7
Constant: #$Mountain .
  isa: TextClassifierClass .

;; #$ComputerLanguage => 7
Constant: #$ComputerLanguage .
  isa: TextClassifierClass .

;; #$SolidFood => 7
Constant: #$SolidFood .
  isa: TextClassifierClass .

;; #$AquaticOrganism => 7
Constant: #$AquaticOrganism .
  isa: TextClassifierClass .

;; #$ReligiousBeliefs => 7
Constant: #$ReligiousBeliefs .
  isa: TextClassifierClass .

;; #$Snake => 7
Constant: #$Snake .
  isa: TextClassifierClass .

;; #$DrugProduct => 7
Constant: #$DrugProduct .
  isa: TextClassifierClass .

;; #$RoadVehicleType => 7
Constant: #$RoadVehicleType .
  isa: TextClassifierClass .

;; #$InfectiousDiseaseType => 6
Constant: #$InfectiousDiseaseType .
  isa: TextClassifierClass .
```

```
;; #$NuclearRelatedMaterial => 6
 Constant: #$NuclearRelatedMaterial .
  isa: TextClassifierClass .

;; #$VenomousAnimal => 6
 Constant: #$VenomousAnimal .
  isa: TextClassifierClass .

;; #$NuclearRelatedThing => 6
 Constant: #$NuclearRelatedThing .
  isa: TextClassifierClass .

;; #$Actor => 6
 Constant: #$Actor .
  isa: TextClassifierClass .

;; #$Currency => 6
 Constant: #$Currency .
  isa: TextClassifierClass .

;; #$Religion => 6
 Constant: #$Religion .
  isa: TextClassifierClass .
;; #$Infection => 6
 Constant: #$Infection .
  isa: TextClassifierClass .

;; #$MedicalDevice => 6
 Constant: #$MedicalDevice .
  isa: TextClassifierClass .

;; #$Bomb => 6
 Constant: #$Bomb .
  isa: TextClassifierClass .

;; #$AlcoholicBeverage => 6
 Constant: #$AlcoholicBeverage .
  isa: TextClassifierClass .

;; #$MigratoryAnimal => 6
 Constant: #$MigratoryAnimal .
  isa: TextClassifierClass .

;; #$DairyProduct => 6
 Constant: #$DairyProduct .
  isa: TextClassifierClass .

;; #$PreparedFood => 6
 Constant: #$PreparedFood .
  isa: TextClassifierClass .

;; #$DiseaseType => 6
 Constant: #$DiseaseType .
  isa: TextClassifierClass .

;; #$Singer => 6
 Constant: #$Singer .
  isa: TextClassifierClass .

;; #$Watercraft-Surface => 6
 Constant: #$Watercraft-Surface .
  isa: TextClassifierClass .

;; #$OverGarment => 6
 Constant: #$OverGarment .
  isa: TextClassifierClass .

;; #$PhysiologicalConditionType => 6
 Constant: #$PhysiologicalConditionType .
  isa: TextClassifierClass .

;; #$Ruminant => 6
 Constant: #$Ruminant .
  isa: TextClassifierClass .

;; #$WomensClothing => 6
 Constant: #$WomensClothing .
  isa: TextClassifierClass .

;; #$InfectionType => 6
 Constant: #$InfectionType .
  isa: TextClassifierClass .

;; #$AilmentCondition => 6
 Constant: #$AilmentCondition .
  isa: TextClassifierClass .

;; #$GermanCar => 6
 Constant: #$GermanCar .
```

```
  isa: TextClassifierClass .

;; #$Watercraft => 6
 Constant: #$Watercraft .
  isa: TextClassifierClass .

;; #$Fish => 6
 Constant: #$Fish .
  isa: TextClassifierClass .

;; ------------------------------
;; Cycorp provided classes
;; ------------------------------
Constant: Organization.
isa: TextClassifierClass.

Constant: MechanicalDevice.
isa: TextClassifierClass.

Constant: SystemOfDevices.
isa: TextClassifierClass.

Constant: Device-SingleUser.
isa: TextClassifierClass.

Constant: Device-OneTimeUse.
isa: TextClassifierClass.

Constant: DeviceWithNoMovingParts.
isa: TextClassifierClass.

Constant: NonPoweredDevice.
isa: TextClassifierClass.

Constant: PoweredDevice.
isa: TextClassifierClass.

Constant: ContainerArtifact.
isa: TextClassifierClass.

Constant: TransportFacility.
isa: TextClassifierClass.

Constant: MilitaryEquipment.
isa: TextClassifierClass.

Constant: VehiclePart.
isa: TextClassifierClass.

Constant: FoodOrDrinkPreparationDevice.
isa: TextClassifierClass.

Constant: MusicalInstrument.
isa: TextClassifierClass.

Constant: InformationStorageDevice.
isa: TextClassifierClass.

Constant: LandTransportationDevice.
isa: TextClassifierClass.

Constant: WaterTransportationDevice.
isa: TextClassifierClass.

Constant: AirTransportationDevice.
isa: TextClassifierClass.

Constant: Tool.
isa: TextClassifierClass.

Constant: ControlDevice.
isa: TextClassifierClass.

Constant: PartOfBuilding.
isa: TextClassifierClass.

Constant: SportsPlayingArea.
isa: TextClassifierClass.

Constant: ConnectedPhysicalPathSystem.
isa: TextClassifierClass.

Constant: OfficialDocument.
isa: TextClassifierClass.

Constant: Graphic-VisualIBT.
isa: TextClassifierClass.

Constant: PublishedConceptualWork.
isa: TextClassifierClass.
```

Constant: Card.
isa: TextClassifierClass.

Constant: AnimalBodyRegion.
isa: TextClassifierClass.

Constant: Sac-Organic.
isa: TextClassifierClass.

Constant: CellPart.
isa: TextClassifierClass.

Constant: ExternalAnatomicalPart.
isa: TextClassifierClass.

Constant: InternalAnatomicalPart.
isa: TextClassifierClass.

Constant: Food.
isa: TextClassifierClass.

Constant: Drink.
isa: TextClassifierClass.

Constant: PlantProduct.
isa: TextClassifierClass.

Constant: Hydrocarbon.
isa: TextClassifierClass.

Constant: Oil.
isa: TextClassifierClass.

Constant: Lipid.
isa: TextClassifierClass.

Constant: OrganicCompound.
isa: TextClassifierClass.

Constant: BiologicalAgentStuff.
isa: TextClassifierClass.

Constant: ToxicSubstance.
isa: TextClassifierClass.

Constant: Cell.
isa: TextClassifierClass.

Constant: Plant.
isa: TextClassifierClass.

Constant: PlantPart.
isa: TextClassifierClass.

Constant: Mammal.
isa: TextClassifierClass.

Constant: Bird.
isa: TextClassifierClass.

Constant: Reptile.
isa: TextClassifierClass.

Constant: Amphibian.
isa: TextClassifierClass.

Constant: Fish.
isa: TextClassifierClass.

Constant: Insect.
isa: TextClassifierClass.

Constant: Person.
isa: TextClassifierClass.

Constant: Movement-Rotation.
isa: TextClassifierClass.

Constant: Translocation.
isa: TextClassifierClass.

Constant: MovementProcess.
isa: TextClassifierClass.

Constant: DangerousActivity.
isa: TextClassifierClass.

Constant: BodyMovementEvent.
isa: TextClassifierClass.

Constant: HumanActivity.
isa: TextClassifierClass.

Constant: BodilyFunctionEvent.
isa: TextClassifierClass.

Constant: TouchingEvent.
isa: TextClassifierClass.

Constant: TransferOfControl.
isa: TextClassifierClass.

Constant: ComputerActivity.
isa: TextClassifierClass.

Constant: ShapeChangeEvent.
isa: TextClassifierClass.

Constant: IBTGeneration.
isa: TextClassifierClass.

Constant: PhysicalCreationEvent.
isa: TextClassifierClass.

Constant: PhysicalDestructionEvent.
isa: TextClassifierClass.

Constant: Configuration.
isa: TextClassifierClass.

Constant: PurposefulAction.
isa: TextClassifierClass.

Constant: ExperiencingEmotion.
isa: TextClassifierClass.

Constant: Perceiving.
isa: TextClassifierClass.

Constant: SocialOccurrence.
isa: TextClassifierClass.

Constant: AilmentCondition.
isa: TextClassifierClass.

Constant: LandTopographicalFeature.
isa: TextClassifierClass.

Constant: BodyOfWater.
isa: TextClassifierClass.

Constant: EcologicalRegion.
isa: TextClassifierClass.

Constant: OutdoorLocation.
isa: TextClassifierClass.

Constant: RuralArea.
isa: TextClassifierClass.

Constant: GeographicalAgent.
isa: TextClassifierClass.

Constant: GeographicalPlace.
isa: TextClassifierClass.

Constant: TacticalTerrainObject.
isa: TextClassifierClass.

Constant: ArtificialMaterial.
isa: TextClassifierClass.

Constant: PartiallyTangibleProduct.
isa: TextClassifierClass.

Constant: FurniturePiece.
isa: TextClassifierClass.

Constant: FlowPath.
isa: TextClassifierClass.

Constant: PathArtifact.
isa: TextClassifierClass.

Constant: DirectedCustomaryPath.
isa: TextClassifierClass.

Constant: Portal.
isa: TextClassifierClass.

Constant: FieldOfStudy.
isa: TextClassifierClass.

Constant: BeliefSystem.
isa: TextClassifierClass.

Constant: Language.
isa: TextClassifierClass.

Constant: WritingSystem.
isa: TextClassifierClass.

Constant: PropositionalConceptualWork.
isa: TextClassifierClass.

Constant: VisualWork.
isa: TextClassifierClass.

Constant: AudioConceptualWork.
isa: TextClassifierClass.

Constant: ComputerFile-CW.
isa: TextClassifierClass.

Constant: SoftwareObject.
isa: TextClassifierClass.

Constant: ConceptualWorkSeries.
isa: TextClassifierClass.

Constant: Sport.
isa: TextClassifierClass.

Constant: Game.
isa: TextClassifierClass.

Constant: GeometricFigure.
isa: TextClassifierClass.

Constant: Surface.
isa: TextClassifierClass.

Constant: Date.
isa: TextClassifierClass.

Constant: LinguisticExpressionType.
isa: TextClassifierClass.

Constant: ConventionalClassificationType.
isa: TextClassifierClass.

Constant: TimeDependentCollection.
isa: TextClassifierClass.

Constant: FirstOrderCollection.
isa: TextClassifierClass.

Constant: Set-Mathematical.
isa: TextClassifierClass.

Constant: CharacterString.
isa: TextClassifierClass.

Constant: PhysicalSeries.
isa: TextClassifierClass.

Constant: EventSeries.
isa: TextClassifierClass.

Constant: CharacterStringToken.
isa: TextClassifierClass.

Constant: PhysicalQuantity.
isa: TextClassifierClass.

Constant: SocialQuantity.
isa: TextClassifierClass.

Constant: MeasurableQuantity.
isa: TextClassifierClass.

Constant: FeelingAttribute.
isa: TextClassifierClass.

Constant: RealNumber.
isa: TextClassifierClass.

Constant: VectorInterval.
isa: TextClassifierClass.

Constant: Predicate.
isa: TextClassifierClass.

Constant: Function-Denotational.
isa: TextClassifierClass.

Constant: Microtheory.
isa: TextClassifierClass.

Constant: ContextualizedInformationStructure.
isa: TextClassifierClass.

Constant: Character-Abstract.
isa: TextClassifierClass.

Constant: LinguisticObject.
isa: TextClassifierClass.

Constant: Card.
isa: TextClassifierClass.

Constant: ElementStuff.
isa: TextClassifierClass.

Constant: Atom.
isa: TextClassifierClass.

Constant: Molecule.
isa: TextClassifierClass.

Constant: Ion.
isa: TextClassifierClass.

Constant: MolecularComponent.
isa: TextClassifierClass.

Constant: OrganicChemicalObject.
isa: TextClassifierClass.

Constant: LandStuff.
isa: TextClassifierClass.

Constant: ComputerCode.
isa: TextClassifierClass.

Constant: SoftwareParameter.
isa: TextClassifierClass.

Constant: ProgramObject.
isa: TextClassifierClass.

Constant: ComputerDataArtifact.
isa: TextClassifierClass.

Constant: ComputerCodeCopy.
isa: TextClassifierClass.

Constant: Gaseous-StateOfMatter.
isa: TextClassifierClass.

Constant: Solid-StateOfMatter.
isa: TextClassifierClass.

Constant: Liquid-StateOfMatter.
isa: TextClassifierClass.

Constant: InorganicCompound.
isa: TextClassifierClass.

Constant: IonicCompound.
isa: TextClassifierClass.

Constant: Electrolyte.
isa: TextClassifierClass.

Constant: Antibiotic.
isa: TextClassifierClass.

Constant: Water.
isa: TextClassifierClass.

Constant: ColoredThing.
isa: TextClassifierClass.

Constant: SymbolicObject.
isa: TextClassifierClass.

Constant: AwardPractice.
isa: TextClassifierClass.

Constant: WeatherEvent.
isa: TextClassifierClass.

Constant: LinearObject.
isa: TextClassifierClass.

Constant: ArrangementOfLikeObjects.
isa: TextClassifierClass.

Constant: StructuredInformationSource.
isa: TextClassifierClass.

Constant: ShapedObject.
isa: TextClassifierClass.

Constant: DispositionalQuantity.
isa: TextClassifierClass.

Constant: DrugSubstance.
isa: TextClassifierClass.

Constant: Nutrient.
isa: TextClassifierClass.

Constant: MolecularStuff.
isa: TextClassifierClass.

Constant:GeometricallyDescribableThing.
isa: TextClassifierClass

APPENDIX C

NOTES ON CLASSIFIER TUNING

Format is : +/- Class name Determination Confidence : Article
So +#$Dog Positive 430.85584987342: Cockapoo
Means the article Cockapoo is a in class +#$Dog with confidence = 430.85584987342
Confidence = | Probility_estimate(+class|Article) - |Probility_estimate (-class |Article)|

No SBPH + raw weighting

+#$Dog Positive 2406.9792795817: Spaniel
+#$Dog Positive 1212.20214021824: List of dog breeds
+#$Dog Positive 798.64949983298: Afghan Hound
+#$Dog Positive 877.647100016839: Beagle
-#$Dog Negative 56.7187709513528: Snoopy
+#$Dog Positive 430.85584987342: Cockapoo
-#$Dog Negative 487.460506437739: Tiger
-#$Dog Negative 2.70286513774136: AI
-#$Dog Negative 55.4547880928585: Garfield
-#$Dog Negative 244.994703293021: Anthrax
-#$Dog Negative 784.994912588489: Coal
-#$Dog Negative 214.966255083011: Sulfur
-#$Dog Negative 1352.76487313717: Japan
-#$Dog Negative 381.63421549031: Cat
-#$Dog Negative 55.2499195145356: Eric Maschwitz
-#$Dog Negative 254.7250606378: Solar nebula
-#$Dog Negative 388.825024720267: William Empson
-#$Dog Negative 18.8798172199176: Radnage
-#$Dog Negative 289.66113353231: Mary Sue
-#$Dog Negative 89.5639005883513: Systemic bias

95% accuracy in classification, with one false negative (Snoopy)
Adding the SBPH did not help

+#$Dog Positive 3806.35378141808: Spaniel
+#$Dog Positive 1995.81561281317: List of dog breeds
+#$Dog Positive 1650.97614270952: Afghan Hound
+#$Dog Positive 2466.31519579585: Beagle
+#$Dog Positive 1224.34760004372: Snoopy
+#$Dog Positive 1204.93487581225: Cockapoo
+#$Dog Positive 2473.75282689498: Tiger
-#$Dog Negative 2.70288371900466: AI
+#$Dog Positive 2365.45909413579: Garfield
+#$Dog Positive 673.273128137676: Anthrax
+#$Dog Positive 1799.14816130907: Coal
+#$Dog Positive 2872.46974424005: Sulfur
+#$Dog Positive 2239.21900462057: Japan
+#$Dog Positive 6553.72425304941: Cat
+#$Dog Positive 172.22759547957: Eric Maschwitz

+#$Dog Positive 968.932893254212: Solar nebula
+#$Dog Positive 2247.69607389084: William Empson
+#$Dog Positive 164.237353384117: Radnage
+#$Dog Positive 1872.74594988555: Mary Sue
+#$Dog Positive 342.197756721471: Systemic bias

Adding Unigrams to the mix returned to the 95% accuracy
It did have a wider range of confidence values

TargetConcept : Dog
Class Result Confidence Article
+#$Dog Positive 3213.12720602032: Spaniel
+#$Dog Positive 3579.17909112299: List of dog breeds
+#$Dog Positive 1354.93599312722: Afghan Hound
+#$Dog Positive 1389.15666632572: Beagle
-#$Dog Negative 445.012687085647: Snoopy
+#$Dog Positive 761.537931248207: Cockapoo
-#$Dog Negative 2197.58104287466: Tiger
-#$Dog Negative 9.27714091842259: AI
-#$Dog Negative 844.211280463001: Garfield
-#$Dog Negative 956.244553748198: Anthrax
-#$Dog Negative 2682.43175352202: Coal
-#$Dog Negative 543.063135618606: Sulfur
-#$Dog Negative 5472.79466465174: Japan
-#$Dog Negative 1145.97389091941: Cat
-#$Dog Negative 228.875600869047: Eric Maschwitz
-#$Dog Negative 836.677899821865: Solar nebula
-#$Dog Negative 1521.65917545886: William Empson
-#$Dog Negative 64.3560446251486: Radnage
-#$Dog Negative 1187.98618021491: Mary Sue
-#$Dog Negative 338.841363790274: Systemic bias

Exponentially Superincreaing Model Weighting (ESMW)
Doesn't change the classification but does improve the certainty margins

+#$Dog Positive 4977.69013466388: Spaniel
+#$Dog Positive 5936.26541240938: List of dog breeds
+#$Dog Positive 2653.84135793543: Afghan Hound
+#$Dog Positive 2687.61676279191: Beagle
-#$Dog Negative 1185.09284216262: Snoopy
+#$Dog Positive 976.323512498762: Cockapoo
-#$Dog Negative 4007.06127218026: Tiger
-#$Dog Negative 8.55286405056205: AI
-#$Dog Negative 2206.20412036468: Garfield
-#$Dog Negative 1581.48467950168: Anthrax
-#$Dog Negative 4647.26800973737: Coal

-#$Dog Negative 1661.08775248475: Sulfur
-#$Dog Negative 9786.41495793476: Japan
-#$Dog Negative 3804.26923608087: Cat
-#$Dog Negative 379.253054956862: Eric Maschwitz
-#$Dog Negative 1542.90919252152: Solar nebula
-#$Dog Negative 2871.38620378904: William Empson
-#$Dog Negative 124.402020243449: Radnage
-#$Dog Negative 2270.38586340209: Mary Sue
-#$Dog Negative 519.901684449083: Systemic bias

Using Exponentially Superincreasing Weights with SBPH using skip characters (3-grams)
Not as good as without SBPH patterns but better than the raw SBPH witho the ESMW

+#$Dog Positive 7185.61956503335: Spaniel
+#$Dog Positive 7665.62347310678: List of dog breeds
+#$Dog Positive 4213.01572139349: Afghan Hound
+#$Dog Positive 4905.71993821619: Beagle
-#$Dog Negative 270.818279708823: Snoopy
+#$Dog Positive 1943.85491359072: Cockapoo
-#$Dog Negative 1808.82430275067: Tiger
-#$Dog Negative 11.3254091670584: AI
-#$Dog Negative 570.811719859543: Garfield
-#$Dog Negative 786.942282959702: Anthrax
-#$Dog Negative 3069.8687671322: Coal
+#$Dog Positive 1025.5751656834: Sulfur
-#$Dog Negative 8006.52113682439: Japan
+#$Dog Positive 2614.23145456804: Cat
-#$Dog Negative 210.287364710504: Eric Maschwitz
-#$Dog Negative 720.626166151196: Solar nebula
-#$Dog Negative 794.015404626: William Empson
+#$Dog Positive 71.6072398171937: Radnage
-#$Dog Negative 523.397701924405: Mary Sue
-#$Dog Negative 213.012038318226: Systemic bias

Removing the "_" as a place holder in the ESMW+SBPH did not change classification from just above however snoopy and sulfur got better

+#$Dog Positive 7284.71774481922: Spaniel
+#$Dog Positive 8308.18062361842: List of dog breeds
+#$Dog Positive 4410.35218078739: Afghan Hound
+#$Dog Positive 5298.20444507478: Beagle
-#$Dog Negative 71.5627784135868: Snoopy
+#$Dog Positive 2133.93709317955: Cockapoo
-#$Dog Negative 1530.09819843437: Tiger
-#$Dog Negative 11.5658702351952: AI
-#$Dog Negative 368.118480317498: Garfield

-#$Dog Negative 819.458345184685: Anthrax
-#$Dog Negative 2891.77718966405: Coal
+#$Dog Positive 951.807831975108: Sulfur
-#$Dog Negative 7887.66396527947: Japan
+#$Dog Positive 3802.14755087637: Cat
-#$Dog Negative 214.311861101585: Eric Maschwitz
-#$Dog Negative 620.055555226601: Solar nebula
-#$Dog Negative 597.339998102078: William Empson
+#$Dog Positive 86.6212628606991: Radnage
-#$Dog Negative 300.322945946886: Mary Sue
-#$Dog Negative 212.477254143047: Systemic bias

Going to 4-grams ESMW + SBPH did not help and turned everything except AI positive

+#$Dog Positive 17495.2658828665: Spaniel
+#$Dog Positive 12394.3830880561: List of dog breeds
+#$Dog Positive 9341.93631705784: Afghan Hound
+#$Dog Positive 11697.4132210876: Beagle
+#$Dog Positive 3568.90254221379: Snoopy
+#$Dog Positive 4701.11059817027: Cockapoo
+#$Dog Positive 6107.80480972456: Tiger
-#$Dog Negative 11.3092090868703: AI
+#$Dog Positive 6005.33382930805: Garfield
+#$Dog Positive 2109.48477062612: Anthrax
+#$Dog Positive 2882.82423403708: Coal
+#$Dog Positive 5466.84047771891: Sulfur
+#$Dog Positive 1833.82737149938: Japan
+#$Dog Positive 21683.6840616742: Cat
+#$Dog Positive 341.46835992983: Eric Maschwitz
+#$Dog Positive 2709.17765108484: Solar nebula
+#$Dog Positive 5874.96369733376: William Empson
+#$Dog Positive 496.152412789714: Radnage
+#$Dog Positive 6197.10377088399: Mary Sue
+#$Dog Positive 1081.83504954152: Systemic bias

However using just 4-grams + ESMW is as good as the previous 3-grams. The numbers are different enough though to show that 4-grams are being hit.

+#$Dog Positive 8291.20423822498: Spaniel
+#$Dog Positive 6590.7070263205: List of dog breeds
+#$Dog Positive 3486.25290953936: Afghan Hound
+#$Dog Positive 3387.75321689753: Beagle
-#$Dog Negative 1868.93242973275: Snoopy
+#$Dog Positive 1164.53329059559: Cockapoo
-#$Dog Negative 5524.7554409974: Tiger
-#$Dog Negative 11.3253773181403: AI

-#$Dog Negative 3666.08853250733: Garfield
-#$Dog Negative 2140.96979656296: Anthrax
-#$Dog Negative 6241.66524311266: Coal
-#$Dog Negative 2726.45808635667: Sulfur
-#$Dog Negative 12870.8349638691: Japan
-#$Dog Negative 6059.65912174882: Cat
-#$Dog Negative 516.482431833527: Eric Maschwitz
-#$Dog Negative 2263.18153604025: Solar nebula
-#$Dog Negative 4031.77785158114: William Empson
-#$Dog Negative 160.59881461002: Radnage
-#$Dog Negative 3200.83339246651: Mary Sue
-#$Dog Negative 732.779345119565: Systemic bias

Adding more dog instances (i.e. as many as can be found in time 30) degrades performance. The system probably needs more negative examples

+#$Dog Positive 6183.09558098327: Spaniel
+#$Dog Positive 10764.1914074048: List of dog breeds
+#$Dog Positive 14686.9358683354: Afghan Hound
+#$Dog Positive 23998.8796173007: Beagle
+#$Dog Positive 768.435109317739: Snoopy
+#$Dog Positive 2134.23574383601: Cockapoo
-#$Dog Negative 616.492641445831: Tiger
-#$Dog Negative 4.61077945304845: AI
+#$Dog Positive 1593.52339137925: Garfield
-#$Dog Negative 443.490189011267: Anthrax
-#$Dog Negative 355.686941447842: Coal
-#$Dog Negative 449.273967931134: Sulfur
-#$Dog Negative 4078.84231482161: Japan
+#$Dog Positive 5382.0973255501: Cat
-#$Dog Negative 180.77406518815: Eric Maschwitz
-#$Dog Negative 602.714386905558: Solar nebula
-#$Dog Negative 1160.01074418399: William Empson
-#$Dog Negative 42.3024638201605: Radnage
+#$Dog Positive 398.806329097133: Mary Sue
-#$Dog Negative 212.509189288339: Systemic bias

Leaving the dog query unlimited and opening the timelimit on the negative to 400 leads to 95% again but Snoopy and Garfield swap places. This means no false negatives, but one false positive.

+#$Dog Positive 6963.62588485729: Spaniel
+#$Dog Positive 11481.3273091466: List of dog breeds
+#$Dog Positive 16194.1399491454: Afghan Hound
+#$Dog Positive 26811.3060019546: Beagle
+#$Dog Positive 67.429962855269: Snoopy
+#$Dog Positive 2056.22608423459: Cockapoo

-#$Dog Negative 3739.00766243204: Tiger
-#$Dog Negative 3.87187181707097: AI
+#$Dog Positive 258.840684516297: Garfield
-#$Dog Negative 1130.98116689513: Anthrax
-#$Dog Negative 2918.94245786345: Coal
-#$Dog Negative 2356.84906405382: Sulfur
-#$Dog Negative 5033.551583856: Japan
-#$Dog Negative 15311.5227146246: Cat
-#$Dog Negative 251.364495348666: Eric Maschwitz
-#$Dog Negative 1449.54479189405: Solar nebula
-#$Dog Negative 2257.59047114255: William Empson
-#$Dog Negative 59.2285546575113: Radnage
-#$Dog Negative 455.082750685018: Mary Sue
-#$Dog Negative 363.561771472316: Systemic bias

Installed a simple cyc query cache system to save time on the generation of the Wikip entries a definite time saver the big issue is what is best to save the wikip list is the shortest, and gives the most benefit for the size.
Saving the url's would probably have a similar compression ratio and be generic enough to handle web info.
Saving the text would offer a lot of speedup but would require compression fro storage
Saving the classifier would be best of all but it would probably be similar to saving the text and would require compression

---

So trying to define Animal One aspect is merging or verifying from the cache
Suppose you had Dog and Animal cached. All those elements in the negative Animal bin are also not dogs. Also the same applies in reverse, all positive Dog examples are also positive Animal examples. So one could merge.
Another aspect of this would be to do a Dynamic programming plan to create the optimal learning set, and then merge as necessary to generate all the lists in the most efficient way possible

---

+#$Animal Positive 324.967782305706: Spaniel
+#$Animal Positive 477.043506081303: List of dog breeds
+#$Animal Positive 756.302544862279: Afghan Hound
+#$Animal Positive 1777.81138670427: Beagle
-#$Animal Negative 731.662603610806: Snoopy
+#$Animal Positive 667.326748757143: Cockapoo
+#$Animal Positive 9287.47494997163: Tiger
-#$Animal Negative 1.37558897408331: AI
-#$Animal Negative 1781.19289102737: Garfield
+#$Animal Positive 115.672356408148: Anthrax
-#$Animal Negative 1336.84874699428: Coal
-#$Animal Negative 7096.92952689924: Sulfur
-#$Animal Negative 1405.21860179614: Japan
+#$Animal Positive 11446.0086092255: Cat

-#$Animal Negative 154.143058859412: Eric Maschwitz
-#$Animal Negative 1387.05553508521: Solar nebula
-#$Animal Negative 900.569423260196: William Empson
-#$Animal Negative 24.3611051947792: Radnage
+#$Animal Positive 602.242463664705: Mary Sue
-#$Animal Negative 26.7822504997239: Systemic bias

The Negative set had no or few people in them.

A simpler query for positivePCWExamples and negativePCWExamples seems to work faster. So its more efficient to run those and quickly filter out the results on the perl side.

Another idea is to add the n-grams immediately from the url to the classifier. This would reduce the strain on the memory requirements, yet has the same processing requirements.

Using random sampling.
What's interesting is how well it does. It can tell an animal article from non-animal so there is a detectible pattern.

+#$Animal Positive 122.82320796018: Spaniel
+#$Animal Positive 1773.19123514148: List of dog breeds
+#$Animal Positive 1222.34055364083: Afghan Hound
+#$Animal Positive 1039.34335787404: Beagle
+#$Animal Positive 911.210541122855: Snoopy
+#$Animal Positive 172.998530190023: Cockapoo
-#$Animal Negative 208.779445772088: Tiger <--
-#$Animal Negative 3.43675702036654: AI
+#$Animal Positive 804.434634217439: Garfield
-#$Animal Negative 1582.83910378531: Anthrax <--
-#$Animal Negative 4813.36324588244: Coal
-#$Animal Negative 7882.70360483674: Sulfur
-#$Animal Negative 2459.02921626304: Japan
-#$Animal Negative 4376.54130048968: Cat <--
+#$Animal Positive 595.220425131964: Eric Maschwitz
-#$Animal Negative 1578.60415292271: Solar nebula
+#$Animal Positive 143.695394901151: William Empson
-#$Animal Negative 45.1233830943129: Radnage
+#$Animal Positive 46.1305106745567: Mary Sue
-#$Animal Negative 403.309341643268: Systemic bias

Another idea for the classifier is to implement it as a hash of hashes. Then one can have multiple classifiers with the data indexed by the class, and one can have do the merging in memory of subsumption classes.

$OmniModel{$bigram}{$classname}
when one performs a classification the system could enumerate the relationship between classes and build the counts dynamically on the fly, as mentioned above.

Another task would be to merge the dmoz/yahoo ontology so each reference page maps to being about some cyc constants. Then all the pages they point to are references for that topic.

What would removal of a stoplist do for performance ???

Adding a stop list improved the results for animals,
stoplist + random sample
Intersting that it included Antrax which is alive but not an animal

+#$Animal Positive 360.562370799544: Spaniel
+#$Animal Positive 3438.39564209848: List of dog breeds
+#$Animal Positive 2213.6814754786: Afghan Hound
+#$Animal Positive 2720.80274329642: Beagle
+#$Animal Positive 913.087392519752: Snoopy
+#$Animal Positive 799.64255662659: Cockapoo
+#$Animal Positive 3361.11491485688: Tiger
-#$Animal Negative 4.19478333382636: AI
+#$Animal Positive 1560.06853288473: Garfield
+#$Animal Positive 159.202419617737: Anthrax <<----
-#$Animal Negative 515.512246021623: Coal
-#$Animal Negative 3311.82417051392: Sulfur
-#$Animal Negative 910.261476426793: Japan
+#$Animal Positive 131109.790124411: Cat
+#$Animal Positive 427.020783509975: Eric Maschwitz
-#$Animal Negative 718.80807914125: Solar nebula
+#$Animal Positive 2025.30300763671: William Empson
+#$Animal Positive 54.8843672092735: Radnage <<-------
+#$Animal Positive 1128.34132347515: Mary Sue
-#$Animal Negative 61.1647985839409: Systemic bias

One cosideration is the use of pronouns for some determinations so it's worth retesting ensuring pronouns are not in the stoplist. In fact in general propositional attachement may imply something about the object under discussion (i.e. what slots are normally filled out for that object).

Also some effects can be due to random sampling.
Changing no parameters, just rerunning to get a different example set returned different results.

+#$Animal Positive 234.935208196697: Spaniel
+#$Animal Positive 1813.81919636781: List of dog breeds
+#$Animal Positive 12760.2546455744: Afghan Hound
+#$Animal Positive 2563.71923381243: Beagle
+#$Animal Positive 1994.05710834726: Snoopy

+#$Animal Positive 491.006958545047: Cockapoo
-#$Animal Negative 179.125829563651: Tiger <<-----
+#$Animal Positive 3.26313818572828: AI
+#$Animal Positive 3774.59567441299: Garfield
-#$Animal Negative 1346.69807105893: Anthrax <<-----
-#$Animal Negative 4454.06699196622: Coal
-#$Animal Negative 8154.40451663826: Sulfur
-#$Animal Negative 1284.07218690636: Japan
-#$Animal Negative 489.483886753209: Cat <<-----
+#$Animal Positive 870.578923776349: Eric Maschwitz
-#$Animal Negative 1412.7984829091: Solar nebula
+#$Animal Positive 2589.89515703148: William Empson
-#$Animal Negative 31.85985180438: Radnage
+#$Animal Positive 2032.1828716915: Mary Sue
-#$Animal Negative 67.3231550849341: Systemic bias

+#$Animal Positive 530.804229421508: Spaniel
+#$Animal Positive 1620.43718495415: List of dog breeds
+#$Animal Positive 1602.44105994287: Afghan Hound
+#$Animal Positive 1404.2043929651: Beagle
+#$Animal Positive 1142.94174666121: Snoopy
+#$Animal Positive 360.411480222996: Cockapoo
+#$Animal Positive 1838.41330362365: Tiger
-#$Animal Negative 1.29704835854233: AI
+#$Animal Positive 1947.66467724039: Garfield
+#$Animal Positive 534.209055379441: Anthrax <<-----
-#$Animal Negative 435.496382703568: Coal
-#$Animal Negative 8049.30632118727: Sulfur
-#$Animal Negative 1274.80000960262: Japan
+#$Animal Positive 6368.88164752547: Cat
+#$Animal Positive 521.408016722904: Eric Maschwitz
-#$Animal Negative 968.492050564673: Solar nebula
+#$Animal Positive 1121.34687681941: William Empson
-#$Animal Negative 30.7587032884958: Radnage
+#$Animal Positive 1845.30363093011: Mary Sue
-#$Animal Negative 65.9898729161687: Systemic bias

+#$Animal Positive 5807.7378244814: Spaniel
+#$Animal Positive 2659.0546611858: List of dog breeds
+#$Animal Positive 2217.84337478715: Afghan Hound
+#$Animal Positive 1997.48584883516: Beagle
+#$Animal Positive 1143.7880214895: Snoopy
+#$Animal Positive 507.483019112296: Cockapoo
+#$Animal Positive 232.966857712658: Tiger
-#$Animal Negative 4.64712870776896: AI

+#$Animal Positive 818.103732067539: Garfield
-#$Animal Negative 1086.64445904728: Anthrax <<-----
-#$Animal Negative 3130.82479363875: Coal
-#$Animal Negative 5281.66758129606: Sulfur
-#$Animal Negative 2247.08386262966: Japan
+#$Animal Positive 862.207839406386: Cat <<-----
+#$Animal Positive 840.502516218752: Eric Maschwitz
-#$Animal Negative 2404.54519635657: Solar nebula
+#$Animal Positive 1660.32649153916: William Empson
+#$Animal Positive 14.1528144651538: Radnage <<-----
+#$Animal Positive 688.370549366809: Mary Sue
-#$Animal Negative 167.164919562896: Systemic bias

Do we need a reference list that is always added on top of the random sample?
So a list is checked against the criteria and placed in the proper category for the example list.
i.e. if you had #$Anthrax on the list it would be checked against the target concept to see if it is an instance. If the target was #$Animal then #$Anthrax would be a negative example.
So this would be a way to create a semantic anchor set in addition to the random sampling.
---
increasing the sampling limits
 my $posTextLimit = 7500000;
 my $negTextLimit = 1500000;

+#$Animal Positive 170.335872447096: Spaniel
+#$Animal Positive 1238.36829006748: List of dog breeds
+#$Animal Positive 2164.23965440328: Afghan Hound
+#$Animal Positive 1741.51638804669: Beagle
+#$Animal Positive 1385.8461223451: Snoopy
+#$Animal Positive 255.112239785296: Cockapoo
+#$Animal Positive 1336.07180821587: Tiger
-#$Animal Negative 2.71832662507424: AI
+#$Animal Positive 2204.86453495891: Garfield
-#$Animal Negative 889.156828442588: Anthrax
-#$Animal Negative 420.001296538656: Coal
-#$Animal Negative 7204.97245945031: Sulfur
-#$Animal Negative 1179.9786258578: Japan
+#$Animal Positive 4025.55322033074: Cat
+#$Animal Positive 567.056096424101: Eric Maschwitz
-#$Animal Negative 2811.06538666843: Solar nebula
+#$Animal Positive 1102.83155630383: William Empson
-#$Animal Negative 12.2810270788759: Radnage
+#$Animal Positive 401.864603278344: Mary Sue
-#$Animal Negative 203.626611366195: Systemic bias

+#$Animal Positive 422.70193621207: Spaniel
+#$Animal Positive 2495.18893823141: List of dog breeds

```
+#$Animal Positive 3021.4580905656: Afghan Hound
+#$Animal Positive 2899.44225331303: Beagle
+#$Animal Positive 2262.72554532674: Snoopy
+#$Animal Positive 761.756842381394: Cockapoo
+#$Animal Positive 2018.465269432: Tiger
+#$Animal Positive 3.43212967098663: AI <<---------
+#$Animal Positive 3734.02474247877: Garfield
-#$Animal Negative 1207.36809293747: Anthrax
-#$Animal Negative 2218.84780315543: Coal
-#$Animal Negative 8843.26675695533: Sulfur
-#$Animal Negative 1169.47422195564: Japan
+#$Animal Positive 4067.49022121506: Cat
+#$Animal Positive 590.019248851062: Eric Maschwitz
-#$Animal Negative 2711.9082137053: Solar nebula
+#$Animal Positive 1978.9317361998: William Empson
+#$Animal Positive 12.2442140189032: Radnage <<---------
+#$Animal Positive 1126.17503677463: Mary Sue
-#$Animal Negative 152.290286038944: Systemic bias

removing "is" and "can" from the stop list

+#$Animal Positive 389.420251631251: Spaniel
+#$Animal Positive 1592.23814759325: List of dog breeds
+#$Animal Positive 1276.79913820569: Afghan Hound
+#$Animal Positive 1505.98690223339: Beagle
+#$Animal Positive 1541.40797361202: Snoopy
+#$Animal Positive 356.259624162427: Cockapoo
+#$Animal Positive 2721.82903780116: Tiger
+#$Animal Positive 4.16328889464617: AI  <<-------------
+#$Animal Positive 865.027291902137: Garfield
-#$Animal Negative 883.451111046757: Anthrax
-#$Animal Negative 2272.7142141154: Coal
-#$Animal Negative 7836.10722898654: Sulfur
-#$Animal Negative 207.896272352664: Japan
+#$Animal Positive 5133.21953649691: Cat
+#$Animal Positive 757.624945523807: Eric Maschwitz
-#$Animal Negative 2141.48949097168: Solar nebula
+#$Animal Positive 1825.52161530228: William Empson
+#$Animal Positive 25.0756068542123: Radnage <<-------------
+#$Animal Positive 1350.29761518084: Mary Sue
+#$Animal Positive 19.4820014890611: Systemic bias

increasing the sampling limits
 my $posTextLimit =  750000;
 my $negTextLimit = 2500000;
```

One could implement a GA where one each bit is wither or not to include an example in the training set. All the ones that are 1 are part of the test set.

Removed the "neutral class"
Added "Tie" files to store the results
Added #$ReferenceExample collection so a cononical set are always used

One could build a GP that generates selection criteria for those elements that form a training set.

APPENDIX D

ARTICLE COUNTS PER CYC CONSTANT

#$Endangered-OrganismTypeByExistentialThreatLevel 11
#$ComputerNetwork 11
#$Bird 11
#$OviparousAnimal 11
#$IBTContentType 11
#$ProtectiveCovering 11
#$SportsOrganization 11
#$InternalOrgan 11
#$CavityWithWalls 11
#$Workplace 11
#$ArtificialMaterial 11
#$Metal 11
#$MusicPerformanceAgent 10
#$HerdAnimal 10
#$Niger-KordofanianLanguageFamily 10
#$Sea 10
#$OuterGarment 10
#$Band-MusicGroup 10
#$Device-UserPowered 10
#$CombustionInstrument 10
#$Device-OneTimeUse 10
#$ConductingMedium 10
#$ElectricalConductor 10
#$FileSharingSystem 10
#$CelestialRegion 10
#$AccountSystem 10
#$JapaneseCar 10
#$MusicPerformanceOrganization 10
#$Musician 10
#$Biology-Topic 10
#$Herbivore 10
#$DeviceWithNoMovingParts 10
#$ComputerHardwareComponent 10
#$HandTool 10
#$Shiny 10
#$ScienceAndNature-Topic 10
#$WildAnimal 9
#$ProtectiveEquipment-Human 9
#$CartographicFeatureType 9
#$UnitedStatesOfAmerica 9
#$TemporalObjectType 9
#$WaterTransportationDevice 9
#$MetallicElementType 9
#$UnitOfMeasureConcept 9
#$OrganSubPartType 9
#$StructuralSupportStuff 9
#$CWInfoStructure 9
#$ComputerMonitor-VideoKind 9
#$ChryslerCar 9
#$GeneralizedTransfer 9
#$LanguagesSpokenInCountries-HormuzArea-Topic 9
#$UpperClass 9
#$EuropeanCitizenOrSubject 9
#$DefenseSystem 9
#$ArtifactualFeatureType 9
#$CelestialObject 9
#$BiochemicallyHarmfulSubstance 9
#$Shoe 9
#$Drink 9
#$Resources 9
#$ProtectiveAttire 9
#$DownwardLocation-Underspecified 9
#$ConsumableProduct 9
#$SheetOfSomeStuff 9
#$CombustibleMaterial 9
#$Constellation 9
#$UnalloyedMetal 9
#$LuxuryCar 9
#$SoccerPlayer 9
#$MilitaryWeaponType 9
#$AgentNowTerminated 8
#$ComputerPeripheralDevice 8
#$CulturalActivity 8
#$TextileArtifact 8
#$GMAutomobile 8
#$Niger-CongoLanguageFamily 8
#$MovementEvent 8
#$Deceased 8
#$DeadLanguage 8
#$FeelingAttribute 8
#$Tuple 8
#$TupleOfIntervals 8
"ORDERED-SET" 8
#$MovementOrShapeChangeEvent 8
#$CommercialOrganization 8
#$SingleUserComputer 8
#$TotallyOrderedScalarIntervalType 8
#$DeskWorker 8
#$PrimitiveOrderedQuantityType 8
#$Fibrous 8
#$ComputerTypeByBrand 8
#$QuantityType 8
#$EmotionalQuantityType 8
#$Translocation 8
#$PersonTypeByPositionInOrg 8
#$InternationalUnitOfMeasure 8
#$AtemporalNecessarilyEssentialCollectionType 8
#$TotallyOrderedCollection 8
#$Lake 8
#$CollectionWithAnEventLikeOrder 8
#$ProjectileLauncher 8
#$Outdoors-ExposedToWeather 8
#$FixedStructure 8
#$InstrumentalArtifact 8
"SET" 8
#$ScalarOrVectorInterval 8
#$Quantity 8
#$TotallyOrderedQuantityType 8
#$ExplosiveDevice 7
#$USUnitOfMeasure 7
#$RecreationalActivity 7
#$ItalicLanguageFamily 7
#$RoyalFamily 7
#$RadiallySymmetricObject 7
#$HumanOccupationConstructObject 7
#$ComputerInterfaceDevice 7
#$Aristocrat 7
#$Monarch-HeadOfState 7
#$NaturalElevation 7
#$Mountain 7
#$ComputerLanguage 7
#$InternationalUnitOfMeasure-Common 7
#$VerticalProtrusion-Topographical 7
#$IntangibleExistingThing 7
#$MountableTransporter 7
#$ProcessType 7
#$MuscleTissue 7

REFERENCES

[Buchanan et al., 1984] Buchanan, B.G. and Shortliffe, E.H., "Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project." Reading, MA: Addison-Wesley, 1984.

[Cavnar et al., 1994] Cavnar, W.B. and Trenkle, J.M., "N-Gram-Based Text Categorization." In Proc. of Third Annual Symposium on Document Analysis and Information Retrieval, Las Vegas, NV, UNLV Publications/Reprographics, 161-175, 1994.

[Chklovski et al., 2004] Chklovski, T. and Pantel, P. "VERBOCEAN: Mining the Web for Fine-Grained Semantic Verb Relations." In Proc. of Conference on Empirical Methods in Natural Language Processing (EMNLP-04), pp. 33-40. Barcelona, Spain, 2004.

[Duthie et al., 2001] Duthie, D.R., Rajesh, D. and Akerkar, R. "Knowledge Representation in KRIS Using Link Grammar Parser." SIBER Research Bulletin 2001.

[Etzioni et al., 2004] Etzioni, O., Cafarella, M., Downey, D., Popescu, A., Shaked, T., Soderland, S., Weld, D. and Yates, A. "Web-scale Information Extraction in KnowItAll." In Proc. of the 13th International Conference on World Wide Web, pp 100-110, New York, NY, 2004.

[Fleischman et al., 2003] Fleischman, M., Hovy, E., Echihabi, A. "Offline Strategies for Online Question Answering: Answering Questions Before They Are Asked." Proceedings of the 41st Meeting of the Association for Computational Linguistics (ACL), Sapporo, Japan. 2003

[Friedland et al., 2004] Friedland, N.S, Allen, P.G., Witbrock, M., Angele, J., Staab, S., Israel, D., Chaudhri, V., Porter, B., Barker, K., and Clark, P., 2004, "Towards a Quantitative, Platform-Independent Analysis of Knowledge Systems," in Proc. of the Ninth International Conference on the Principles of Knowledge Representation and Reasoning (KR 2004), Whistler, 507-515.

[Guha, 1991] Guha, R.V. "Contexts: A Formalization and Some Applications." Ph.D. Thesis, Stanford University, STAN-CS-91-1399-Thesis, 1991.

[Haase, 1986] Haase, K. "Discovery Systems." In Proc. of ECAI-86, 1986.

[Hovy et al., 2002a] Hovy, E., Hermjakob, U., Ravichandran, D. "A Question/Answer Typology with Surface Text Patterns." In Proceedings of the DARPA Human Language Technology Conference (HLT), San Diego, CA, 2002.

[Hovy et al., 2002b] Hovy, E., Hermjakob, U., Lin, C-Y., Ravichandran, D. "Using knowledge to facilitate factoid answer pinpointing Full text." In Proc. of the 19th International Conference on Computational Linguistics, Vol. 1, Taipei, Taiwan, 2002.

[Lenat, 1976] Lenat, D.B. "AM: An Artificial Intelligence Approach to Discovery in Mathematics as Heuristic Search." Ph.D. Dissertation, Stanford University, STAN-CS- 76-570, 1976.

[Lenat et al., 1983] Lenat, D.B., Borning, A., McDonald, D., Taylor, C., Weyer, S. "Knoesphere: Building Expert Systems with Encyclopedic Knowledge." In Proc. of the 8th International Joint Conference on Artificial Intelligence, Vol 1, pp 167–169, Karlsruhe, Germany, August 1983.

[Lenat et al., 1984] Lenat, D. and Brown, J. 1984. "Why AM and EURISKO appear to work." Artificial Intelligence, 23, No. 3, pp. 269-294.

[Lenat, 1982] Lenat, D.B. "AM: Discovery in Mathematics as Heuristic Search." McGraw-Hill, New York, NY (1982) 1–225.

[Lenat, 1995] Lenat, D.B., 1995, "Cyc: A Large-Scale Investment in Knowledge Infrastructure." Communications of the ACM 38, no. 11.

[Lewis et al., 1994] Lewis, D.D., & Gale, W.A. "A sequential algorithm for training text classifiers." Proc. of SIGIR-94, 17th ACM International Conference on Research and Development in Information Retrieval, pp. 3-12. Dublin, IE: Springer-Verlag.

[Lin et al., 2001] Lin ,D. and Pantel, P. "DIRT-Discovery of Inference Rules from Text." In Proceedings of ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD-01), pp. 323-328, San Francisco, CA.

[Lindsay, 1980] Lindsay, R.K., Buchanan B.G., Feigenbaum, E.A. and Lederberg, J., 1980. "Application of Artificial Intelligence for Chemistry: The DENDRAL Project." New York: McGraw-Hill, 1980.

[Livingston, 2001] Livingston, G. R. "A Framework for Autonomous Knowledge Discovery from Databases." Ph.D. Dissertation, Dept. of Computer Science, Univ. of Pittsburgh, Pittsburgh, PA.

[Matthews et al., 2003] Matthews, G. and Vizedom, A. "An Interactive Dialogue System for Knowledge Acquisition in Cyc." In Proceedings of the IJCAI-03 Workshop on Mixed-Initiative Intelligent Systems, Acapulco, Mexico, 2003, pp 138-145.

[Matuszek et al., 2005] Matuszek, C., Witbrock, M., Kahlert, R.C., Cabral, J., Schneider, D., Shah, P., and Lenat, D. "Searching for Common Sense: Populating Cyc from the Web." In Proc. of the 20th National Conference on Artificial Intelligence, Pittsburgh, PA, July 2005.

[Miller et al., 1990] Miller, G.A., Beckwith, R., Fellbaum, C., Gross, D., and Miller, K.J. "Introduction to WordNet: An on-line lexical database." Journal of Lexicography, 3(4), 235-244. 1990.

[Mladenic et al., 2004] Mladenic, D. and Grobelnik, M. "Mapping documents onto web page ontology." Webmining: from web to semantic web: EWMF 2003, Springer Lecture Notes 2004.

[Molla-Aliod et al., 2002] Molla-Aliod, D. and Hutchinson, B., "Dependency-based semantic interpretation for answer extraction." In 2002 Australasian Natural Language Processing Workshop.

[Panton et al., 2002] Panton, K., Miraglia, P., Salay, N., Kahlert, R.C., Baxter, D., and Reagan, R., 2002, "Knowledge Formation and Dialogue using the KRAKEN Toolset," in Proceedings of the Fourteenth Innovative Applications of Artificial Intelligence Conference (IAAI-2002), pp 900-905.

[Pantel et al., 2004a] Pantel, P., Ravichandran, D. and Hovy, E. "Towards Terascale Knowledge Acquisition." In Proc. of Conference on Computational Linguistics (COLING-04), pp. 771-777, Geneva, Switzerland.

[Pantel et al., 2004b] Pantel, P., Ravichandran, D. and Hovy, E. "The Terascale Challenge." In Proc. of KDD Workshop on Mining for and from the Semantic Web (MSW-04), pp. 1-11, Seattle, WA.

[Ramachandran, 2005] Ramachandran, D., Reagan, P., and Goolsbey, K. "First-Orderized ResearchCyc: Expressivity and Efficiency in a Common-Sense

Ontology." In Papers from the AAAI Workshop on Contexts and Ontologies: Theory, Practice and Applications. Pittsburgh, PA, July 2005.

[Ravichandran et al., 2002] Ravichandran, D. and Hovy, E. "Learning Surface Text Patterns for a Question Answering system." In Proceedings of the 40th ACL Conference, Philadelphia, PA.

[Schubert, 2002] Schubert, Lenhart K. "Can we derive general world knowledge from texts?" Human Language Technology Conference (HLT 2002), San Diego, CA, March 24-27, 2002, pp. 94-97.

[Schubert, 2003] Schubert, Lenhart K. "Deriving General World Knowledge from Texts and Taxonomies." From http://www.cs.rochester.edu/~schubert/projects/world-knowledge-mining.html

[Singh et al., 2002] Singh, P., Lin, T., Mueller, E., Lim, G., Perkins, T., and Zhu, W.L., 2002, "Open Mind Common Sense: Knowledge Acquisition from the General Public." Proceedings of the First International Conference on Ontologies, Databases, and Applications of Semantics for Large Scale Information Systems, Irvine, CA.

[Sleator et al., 1991] Sleator, D. and Temperley, D. 1991. "Parsing English with a link grammar." Technical Report CMU-CS-91-196, Department of Computer Science, Carnegie-Mellon University.

[Sviokla, 1990] Sviokla, J.J. "An examination of the impact of expert systems on the firm: the case of XCON." In MIS Quarterly, V. 14, no. 2, pp 127-140. June 1990.

[Witbrock et al., 2003] Witbrock, M., Baxter, D., Curtis, J., Schneider, D., Kahlert, R.C., Miraglia, P., Wagner, P., Panton, K., Matthews, G., and Vizedom, A. "An Interactive Dialogue System for Knowledge Acquisition in Cyc." In Proc. of the 18th International Joint Conference on Artificial Intelligence, Acapulco, Mexico, 2003.

[Witbrock et al., 2005] Witbrock, M., Matuszek, C., Brusseau, A., Kahlert, R.C., Fraser, C.B., and Lenat, D., "Knowledge Begets Knowledge: Steps towards Assisted Knowledge Acquisition in Cyc." In Proc. of the AAAI 2005 Spring Symposium on Knowledge Collection from Volunteer Contributors, Stanford, CA, March 2005.

[Yerazunis, 2003] Yerazunis, W.S., "Sparse Binary Polynomial Hashing and the CRM-114 Discriminator," 2003, Mitsubishi Electric Research Laboratories http://crm114.sourceforge.net/CRM114_paper.html

[Yerazunis, 2004] Yerazunis, W.S., "The Spam Filtering Plateau at 99.9% Accuracy and How to Get Past It," 2004, Mitsubishi Electric Research Laboratories (MERL), MIT Spam Conference, 2004.. http://www.merl.com/reports/docs/TR2004-091.pdf

[Yerazunis et al., 2005] Yerazunis, W.S., Chabra, S., Siefkes, C., Assis, F., and Gunopulos, D., "A Unified Model of Spam Filtration," MIT Spam Conference, January 2005.

[Zhang et al., 2001] Zhang, Lei & Yu, Yong, "Learning to Generate CGs from Domain Specific Sentences," in the Proceedings of the 9th International Conference on Conceptual Structures, LNAI 2120, July 30-August 3, 2001, Stanford, CA, USA.