Semi-Automatic Domain Ontology Creation from Text Resources

Mithun Balakrishna, Dan Moldovan, Marta Tatu, Marian Olteanu

Lymba Corporation
Richardson TX 75080 USA
mithun@lymba.com, moldovan@lymba.com, marta@lymba.com, marian@lymba.com

Abstract

Analysts in various domains, especially intelligence and financial, have to constantly extract useful knowledge from large amounts of unstructured or semi-structured data. Keyword-based search, faceted search, question-answering, etc. are some of the automated methodologies that have been used to help analysts in their tasks. General-purpose and domain-specific ontologies have been proposed to help these automated methods in organizing data and providing access to useful information. However, problems in ontology creation and maintenance have resulted in expensive procedures for expanding/maintaining the ontology library available to support the growing and evolving needs of analysts. In this paper, we present a generalized and improved procedure to automatically extract deep semantic information from text resources and rapidly create semantically-rich domain ontologies while keeping the manual intervention to a minimum. We also present evaluation results for the intelligence and financial ontology libraries, semi-automatically created by our proposed methodologies using freely-available textual resources from the Web.

1. Introduction

Over the past decade, ontologies and knowledge bases (KBs) have gained popularity due to their high potential benefits in a number of applications including data/knowledge organization and search applications (Cimiano, 2006). (Moldovan et al., 2007) presented a methodology for integrating domain and general-purpose ontologies into a question-answering and faceted-search system to help intelligence analysts in organizing data and accessing useful information. Though this ontology integration is beneficial, it is very well known that ontology creation is an expensive process (Ratsch et al., 2003; Pinto and Martins, 2004) and hence was referred to as the *knowledge acquisition bottleneck* in (Cimiano, 2006). The modeling of non-trivial domain ontologies is difficult, and is time and resource intensive (Cimiano, 2006). The *knowledge acquisition bottleneck* problems in ontology creation and maintenance have resulted in expensive procedures for maintaining and expanding the ontology library available to support the growing and evolving needs of analysts in various domains.

(Balakrishna and Srikanth, 2008) presented an ontology modeling methodology for the National Intelligence Priorities Framework (NIPF) (FBI, 2009) topics using Jaguar-KAT (Moldovan and Girju, 2001; Moldovan et al., 2007), a state-of-the-art tool for knowledge acquisition and modeling. In this paper, we present a generalized and improved procedure to semi-automatically create domain ontologies from textual resources while keeping manual intervention to the minimum. We first present the methodology used in our Polaris tool to automatically extract deep semantic information from text. We then present the generalized, semi-automatic domain-ontology modeling algorithm built into our Jaguar tool. We use our generalized domain-ontology development (creation and maintenance) methodology to create ontology libraries for 40 intelligence topics (including NIPF topics) and 10 topics from the financial domain. Following the ontology evaluation levels defined in (Brank et al., 2005), we present detailed evaluations focused on the *Lexical, Vocabulary, or Data Layer* level and the *Other Semantic Relations* level. Polaris and Jaguar are the key components in Lymba's knowledge extraction and representation platform (K-Platform).

2. Polaris - Automatic Semantic Relation Extraction from Text

Polaris, a semantic parser, automatically extracts deep semantic information from text. Polaris is based on a set of 26 semantic relations which Lymba has defined. Semantic relations are abstractions of underlying relations between concepts, and can occur within a word, between words, between phrases, and between sentences (Badulescu and Moldovan, 2008). Semantic relations are useful because they provide connectivity between concepts and contexts. Also, detecting and extracting semantic relations are essential steps toward the ultimate goal of machine text understanding. Semantic relations allow for richer ontologies and knowledge bases which can capture contextual knowledge, events, and firmer assertions. Lymba's set of 26 semantic relations for text understanding is summarized in Table 1. These 26 relations have been carefully selected for their usefulness in Natural Language Processing (NLP), for the feasibility of their automatic extraction from text, and for the broadest semantic coverage with the least amount of overlap. They cover most of the thematic roles proposed by Fillmore and others, and the semantic roles in PropBank. Our goal is to cover as much semantics as possible with as few relations as possible.

In the sample sentence *He carefully disarmed the letter bomb*, the compound nominal *letter bomb* alone contains at least 4 semantic relations: *letter bomb* IS-A *bomb*, *letter bomb* IS-A *letter*, *bomb* is AT-LOCATION *letter*, and *bombing* is the PURPOSE of *letter bomb*. The sentence also includes several other semantic relations: *He* is the AGENT of *disarm*, *carefully* is the MANNER of *disarmed*, and the *letter bomb* is the THEME (or object) of *disarmed*. Together, these semantic relations can give a structured picture of the specified event: who was involved, what was

Relation	Definition	Code	Relation	Definition	Code
Agent(X,Y)	X is the agent of Y; X is prototypically a person	AGT	Association(X,Y)	Person X is associated with Person Y; the relation is not necessarily kinship	ASO
At-Location(X,Y)	X is at-location Y or where X takes place	AT-L	At-Time(X,Y)	X is at-time Y or when X takes place	AT-T
Cause(X,Y)	X causes Y	CAU	Experiencer(X,Y)	X is an experiencer of Y; involves cognition and senses	EXP
Influence(X,Y)	X caused something to happen to Y	IFL	Instrument(X,Y)	X is an instrument in Y	INS
Intent(X,Y)	X is the intent/goal/reason of Y	INT	IS-A(X,Y)	X is a (kind of) Y	ISA
Justification(X,Y)	X is the reason or motivation or justification for Y	JST	Kinship(X,Y)	X is a kin of Y; X is related to Y by blood or by marriage	KIN
Make(X,Y)	X makes Y	MAK	Manner(X,Y)	X is the manner in which Y happens	MNR
Part-Whole(X,Y)	X is a part of Y	PW	Possession(X,Y)	X is a possession of Y; Y owns/has X	POS
Property(X,Y)	X is a property/attribute/value of Y	PRO	Purpose(X,Y)	X is the purpose for Y	PRP
Quantification(X,Y)	X is a quantification of Y; Y can be an entity or event	QNT	Recipient(X,Y)	X is the recipient of Y; X is an animated entity.	RCP
Source(X,Y)	X is the source, origin or previous location of Y	SRC	Stimulus(X,Y)	X is the stimulus of Y; Perceived through senses	STI
Synonymy(X,Y)	X is a synonym/name/equal for/to Y	SYN	Theme(X,Y)	X is the theme of Y	THM
Topic(X,Y)	X is the topic/focus of cognitive communication Y	TPC	Value(X,Y)	X is the value of Y	VAL

Table 1: The set of 26 semantic relations used in Polaris.

done, and to whom; and for what purpose.

2.1. Syntactic Patterns

To find semantic relations in text, Polaris uses a combination of state-of-the-art text processing, pattern matching and machine learning techniques. In the first step, low-level NLP processing, such as part-of-speech tagging, named entity recognition, syntactic parsing and word sense disambiguation, co-reference resolution, are used to structure the text. The parse tree is then broken down into a number of syntactic patterns that Polaris can analyze. These syntactic patterns include verbs and their arguments, complex nominals, adjective phrases, adjective clauses, and others. There are six primary pattern types discovered within noun phrases: N-N and Adj-N (which comprise compound nominals), 's and of (Genitive patterns), Adjective Phrases, and Adjective Clauses. The first five are further subdivided into nominalized and non-nominalized occurrences, giving a total of 11 patterns discovered within compound nominals. The training corpus source for the noun phrase patterns is Wall Street Journal (TreeBank 2), L.A. Times (TREC 9), and XWN 2.0 (Harabagiu and Moldovan, 1998). There are also five verb argument level patterns being discovered: NP verb, verb NP, verb PP, verb ADVP, and verb S. The training corpus source for the verb argument patterns is FrameNet (Baker et al., 1998).

2.2. Machine Learning Classifiers

Polaris next runs classifiers on each section of text that matched a syntactic pattern. The classifiers examine features of the text and attempt to determine whether any of the 26 relations apply between the elements of the pattern. Most of the classifiers are based on one of four different machine learning algorithms: Decision Trees, Naive Bayes, Support Vector Machine (SVM), and Semantic Scattering (a new learning algorithm that uses WordNet classes to find the most probable relation that holds between two nouns (Badulescu and Moldovan, 2008)). Some of these machine-learning classifiers use a per-relation approach to output only one specific relation they were trained to recognize, while others use a per-pattern approach which could potentially output any of the 26 semantic relations. Additionally, some classifiers containing human-coded rules are used for the most explicit and unambiguous cases. These three methods form a hybrid approach which produces better results than any one approach on its own.

3. Jaguar - Domain Ontology Generation

Jaguar processes textual resources and rapidly builds domain-specific ontologies in Lymba's proprietary format or in standard formats like W3C's RDF and OWL. The text input to Jaguar can come from a variety of document sources, including Text, MS Word, PDF and HTML web pages, etc. A Jaguar knowledge-base includes the following constituents:

- Ontological Concepts: basic building blocks of an ontology
- Hierarchy: structure that captures universal knowledge on certain ontological concepts via transitive relations (e.g. ISA, Part-Whole, Locative, etc)

- Contextual Knowledge: knowledge clusters that capture non-universal and contextual knowledge via all semantic relations discovered by the semantic parser
- Axioms on Demand: captures assertions about concepts of interest generated from the available knowledge and is useful for reasoning on text

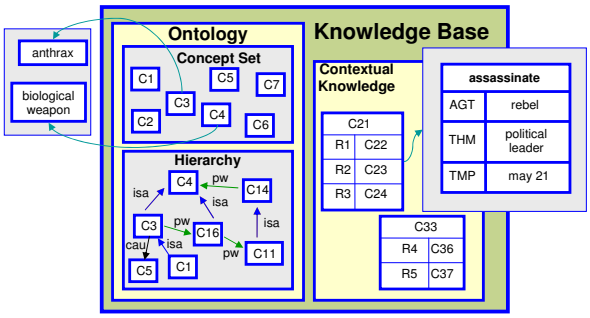

Figure 1: An example Jaguar knowledge-base containing concepts, hierarchy and contextual knowledge.

Note that we consider a knowledge base to contain ontological (universal) knowledge plus contextual knowledge. Figure 1 shows an example Jaguar knowledge-base containing concepts, hierarchy and contextual knowledge. Our domain-ontology modeling algorithm in Jaguar is divided into the following steps:

- Document Pre-Processing
- Concept and Relation Discovery
- Knowledge Classification or Hierarchy Formation

3.1. Document Pre-Processing

The input to Jaguar includes a document collection, and a seeds file containing the concepts/keywords of interest in the domain. The first step in Jaguar involves the processing of the document collection, and seeds file augmentation.

3.1.1. Extracting Textual Content

We first extract text from the input document collection and then filter/clean-up the extracted text. The textual input to Jaguar comes from all possible document types, including MS Word, PDF and HTML web pages, and is therefore prone to having many irregularities such as incomplete, strangely formatted sentences, headings, and tabular information. The text extraction and filtering rules include, conversion or removal of non-ASCII characters, verbalization of infoboxes and tables, conversion of punctuation symbols, among others.

3.1.2. Seeds Set Selection

For each topic being modeled, we define a seed set containing the concepts or keywords of interest in the topic. This seed set is used to determine the set of text sentences of interest in a topic's document collection. We used an algorithm based on TF-IDF, table-of-content rules, document title/sub-title processing, and table and figure caption processing to automatically augment the initial set of seeds that we manually selected. We then manually filter through the augmented seeds set to create the final seeds set. We created seeds sets containing on average of 47 concepts of interest for each topic defined in our intelligence and financial domain ontology libraries.

3.2. Concept and Relation Discovery

For each topic, the extracted text files are processed through a set of state-of-the-art NLP tools: part-of-speech tagging, named-entity recognition, syntactic parsing, word-sense disambiguation, co-reference resolution, and semantic parsing (or semantic relation discovery). The concept discovery module then extracts the concepts of interest using the concepts defined in the input seeds set as a starting point and growing it based on the NLP information extracted from the input text collection. For our current ontology creation experiment, we focus only on noun concepts and their semantic relations. Figure 2 depicts this iterative process of extracting domain-specific concepts and semantic relations using seed concepts.

The concept discovery module first identifies sentences in the document which contain the seed concepts. It then analyzes the syntactic parse tree of each such sentence and identifies Noun Phrases (NPs) containing or related to the topic seed concepts. Every NP is considered to be a potential new concept. Such NP is then processed to extract well-formed noun concepts using syntactic patterns and rules:

- Collocations: search the NP for word collocations that are defined in WordNet as a concept. Thus, *checking account* is extracted as a concept as shown in the example in Figure 2.
- Named Entities: search the NP for named-entities and extract them as concepts.
- Descriptive Adjective Filtering: when adjectives are part of the NPs, extract as concepts only those NPs that are formed with relational and participial adjectives while the NPs with descriptive adjectives are discarded since descriptive adjectives do not add important information to the nouns that they modify. Hence, concepts like *british tea* (relational adjective based) and *boiling water* (participial adjective based) are extracted while concepts like *fast growth* and *high interest* (descriptive adjective based) are discarded.
- Determiner and Numeral Filtering: search the NP and prevent the determiner/numeral nodes from being part of any concept under that NP.
- Concept Splitting: if a conjunction or some *concept-delimiting* punctuation like "," or ":" is found under an NP, split the NP to create two concepts at the point of the conjunction or punctuation.

Noun concepts that are part of the seed set, their semantic relations (extracted from the semantic parser, Polaris (Bixler et al., 2005; Badulescu and Moldovan, 2008)), and the noun concepts involved in semantic relations with seed concepts are marked for further processing. The selected concepts and semantic relations are then processed

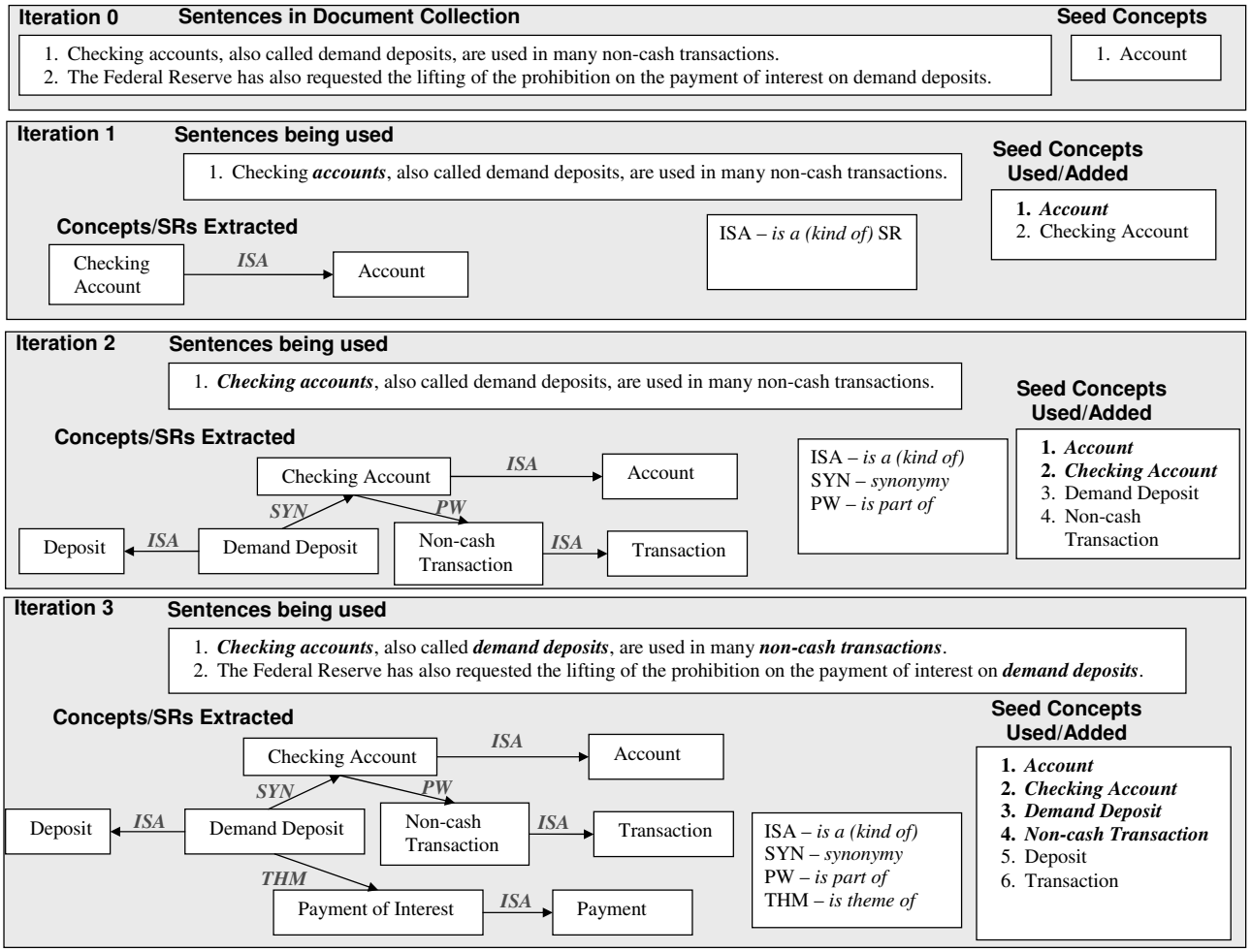

Figure 2: Example depicting the iterative process of extracting domain concepts and semantic relations using seeds.

and used to populate one or many semantic contexts, groups of relations or nested contexts which hold true around a common central concept. The seed set is then augmented with concepts that have hierarchical relations with the target words or seeds. While processing the intelligence and financial topic collections, we used ISA, Part-Whole and Synonymy relations as hierarchical relations required for automatically augmenting the seeds concept set.

The entire process of sentence selection, concept extraction, semantic relation extraction, and seed concepts set augmentation is repeated in an iterative manner, n number of times (by default, $n = 3$).

3.3. Knowledge Classification

Using the discovered concepts and semantic relations, the knowledge classification module forms a hierarchical structure within the set of identified domain concepts via transitive semantic relations that generally hold to be universally true (e.g. ISA, Part-Whole, Locative, etc). The classification is based on the subsumption principle (Schmolze and Lipkis, 1983; Woods, 1991; Baader et al., 2003). The knowledge classification module builds the domain-specific hierarchical structure using WordNet as the upper ontology and extending it using the concepts and semantic relations discovered in the text. Certain hypernymns discovered during classification contain anomalies (causing cycles) or redundancies. Hence, we run them through a conflict resolution engine to detect and correct inconsistencies. The knowledge classification module creates the domain hierarchy link by link (semantic relation by semantic relation) and follows a conflict avoidance technique, wherein each new link is tested for causing inconsistencies before being added to the hierarchy.

The knowledge classification module creates the ontology hierarchy by performing the following steps:

- *Step 1*: From the discovered set of semantic relations, lets consider all the IS-A relations. There are two distinct possibilities:
  - A IS-A relation links a WordNet concept with another concept *c* extracted from the text. The concept *c* is linked to WordNet and added to the hierarchy.
  - A hypernymy relation links a seed concept with a non-seed concept found in the text. Such non-seed concepts are added to the hierarchy but they form some isolated islands since are not yet linked to the main hierarchical tree.
- *Step 2*: Using the hierarchy forest obtained in Step 1,

run the following procedures on concepts that do not link to WordNet directly or indirectly:

Procedure 1: Classify a concept of the form [$word$, $head$] with respect to concept [$head$]. Here, we consider only those head nouns/adjectives that do not have any hyponyms. The more complex case when the head has other concepts under it is treated by *Procedure 4*. The classification is based on the simple idea that a compound concept [$word$, $head$] is ontologically subsumed by concept [$head$]. For example, *checking account* is a kind of *account*, thus linked by a relation *hypernymy(account, checking account)*.

Procedure 2: Classify a concept [$word_1$, $head_1$] with respect to another concept [$word_2$, $head_2$]. If $head_1$ subsumes $head_2$ and $word_1$ subsumes $word_2$, then [$word_1$, $head_1$] subsumes [$word_2$, $head_2$]. E.g. [$Asian\ country$] subsumes [$Japan$] and [$interest\ rate$] subsumes [$discount\ rate$] and hence concept [$Asian\ country\ interest\ rate$] subsumes concept [$Japan\ discount\ rate$].

Note that the subsumption may not always be a direct and may consist of a chain of subsumption relations since subsumption is (usually) a transitive relation. If there is no direct subsumption relation in WordNet between $word_1$ and $word_2$, and/or $head_1$ and $head_2$, but there are common subsuming concepts. Then, we pick the Most Specific Common Subsumer (MSCS) concepts of $word_1$ and $word_2$, and of $head_1$ and $head_2$, respectively. Then form a concept [MSCS($word_1$,$word_2$), MSCS($head_1$, $head_2$)] and place [$word_1$, $head_1$] and [$word_2$, $head_2$] under it. E.g. to classify [$Japan\ discount\ rate$] with respect to [$Germany\ prime\ interest\ rate$], we add the MSCS concept [$country\ interest\ rate$] to the hierarchy and place both the concepts [$Japan\ discount\ rate$] and [$Germany\ prime\ interest\ rate$] under it.

Procedure 3: To classify a concept [$word_1$, $word_2$, $head$]:

1. If there is already a concept [$word_2$, $head$] in the knowledge base under [$head$], then place [$word_1$, $word_2$, $head$] under concept [$word_2$, $head$].
2. If there is already a concept [$word_1$, $head$] in the knowledge base under [$head$], then place [$word_1$, $word_2$, $head$] under concept [$word_1$, $head$].
3. If both 1 and 2 are true then place [$word_1$, $word_2$, $head$] under both concepts [$word_2$, $head$] and [$word_1$, $head$].

Procedure 4: Classify a concept [$word_1$, $head$] with respect to a concept hierarchy under [$head$]. The task here is to identify the Most Specific Subsumer (MSS) from all the concepts under the head that subsumes [$word_1$, $head$]. By default, [$word_1$, $head$] is placed under [$head$], however, since it may be more specific than other hyponyms of [$head$], a more complex classification analysis needs to be implemented.

We identify the set of semantic relations into which the verbs used in the WordNet gloss definitions are mapped into for the purpose of working with a manageable set of relations that may describe the concepts restrictions. In WordNet these basic relations are already identified and it is easy to map every verb into such a semantic relation. For the newly discovered concepts, their defining relations need to be retrieved from texts. Human assistance is required to pinpoint the most characteristic relations that define a concept.

Let AR_aC_a and BR_bC_b denote the relationships that define concepts A and B respectively. The following is the algorithm for the relative classification of two concepts A and B:

- Extract relations (denoted by verbs) between concept and other gloss concepts. E.g. $AR_{a1}C_{a1}$, $AR_{a2}C_{a2}$,....$AR_{am}C_{am}$; $BR_{b1}C_{b1}$, $BR_{b2}C_{b2}$,....$BR_{bn}C_{bn}$
- A subsumes B if and only if:
  - Relations R_{ai} subsume R_{bi}, for $1 \leq i \leq m$.
  - C_{ai} subsumes or is a meronym of C_{bi}.
  - Concept B has more relations than concept A, i.e. $m \leq n$

Procedure 5: Merge a structure of concepts with the rest of the knowledge base following a conflict resolution technique. It is possible that structures consisting of several inter-connected concepts are formed in isolation of the main hierarchy as a result of some procedures. We merge such structures with the main hierarchy such that the new ontology will be consistent and does not contain anomalies (causing cycles) or redundancies. We bridge whenever possible the structure concepts and the main hierarchy concepts while destroying some hypernymy relations to keep the consistency.

- *Step 3*: Repeat *Step 2* for all the concepts that do not link to WordNet several times till no more changes occur. This reclassification is necessary since the insertion of a concept into the hierarchy may perturb the ordering of other surrounding concepts in the hierarchy.
- *Step 4*: Add the remaining relation types other than the IS-A type to the new knowledge base. The IS-A relations have already been used in the hierarchy formation, but the other relation types e.g. Cause, Part-Whole, Influence, etc. need to be added to the knowledge base.

3.4. Ontology Merging

Ontology merging is useful for systems where small chunks of the input text are processed at different parts of the system or at different times, and then subsequently merged (Choi et al., 2006; Gal and Shvaiko, 2009). Jaguar provides an ontology maintenance option to layer ontologies from many different runs. Figure 3 depicts the process of merging two ontologies through conflict resolution algorithms. We perform merging by enumerating the concepts

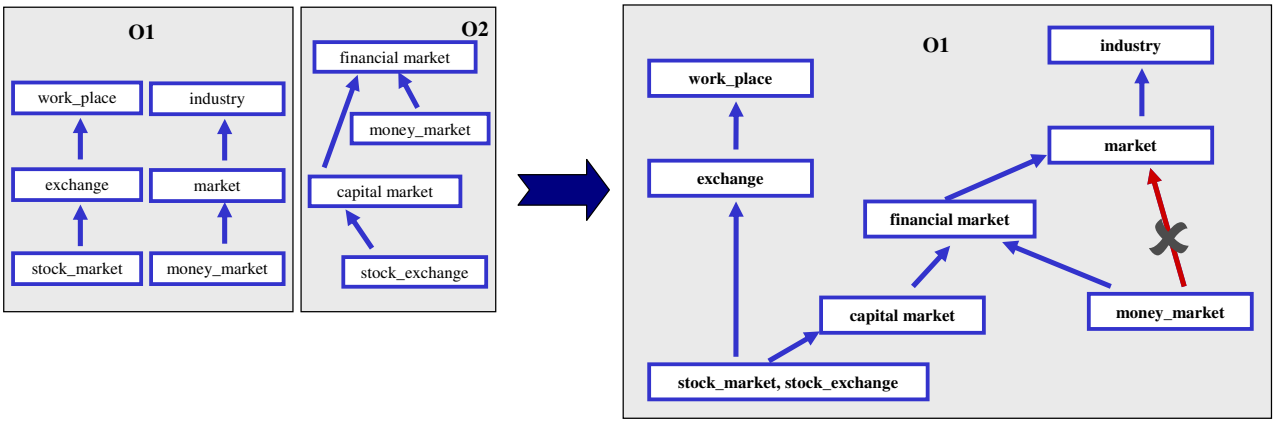

Figure 3: An example depicting Jaguar's merging of two ontologies through conflict resolution algorithms.

Semantic Relation	Definition	Example	Code
ISA	X is a (kind of) Y	[XY] [John] is a [person]	ISA
Part-Whole/Meronymy	X is a part of Y	[XY] [The engine] is the most important part of [the car] [XY] [steel][cage] [YX] [faculty] [professor] [XY] [door] of the [car]	PW
Cause	X causes Y	[XY] [Drinking] causes [accidents]	CAU

Table 2: Subset of semantic relations used to evaluate Jaguar's automatic domain-ontology generation from text.

and relations in the smaller ontology O_2 and adding them to the larger or reference ontology O_1. Jaguar performs the following steps:

- We first merge O_2's concept set into O_1's concept set. If a concept c_1 from O_1 exists in O_2 with the same lexical signature then c_1 is ignored. We use WordNet synsets as a reference to group concepts with different lexical signatures as a single concept. E.g. *stock_market* and *stock_exchange* represent the same concept but have different lexical signatures.
- We then add non-hierarchical semantic clusters from O_2 into O_1. Since the two semantic clusters are independent of each other, this merge is simple and direct.
- Finally, the hierarchical relations from O_2 are added into O_1. If a relation from O_2 already exists in O_1, then it is ignored else we add the relation to O_1 and run the five classification procedures described in Step 2 of the knowledge classification algorithm from Section 3.3..

4. Results

4.1. Semantic Relation Evaluation

Sections 2.1. and 2.2. presented details regarding the syntactic patterns and machine learning algorithms used by Polaris to discover semantic relations in text. The training corpus source for the noun phrase patterns is Wall Street Journal (TreeBank 2), L.A. Times (TREC 9), and XWN 2.0 (Harabagiu and Moldovan, 1998). The training corpus source for the verb argument patterns is FrameNet (Baker et al., 1998).

Lymba has created the following three evaluation corpora to benchmark the semantic relations extracted by the Polaris system:

- TreeBank: we manually annotated 500 random sentences from the Penn Treebank 3 corpus with 5879 semantic relations.
- GlassBox Human: 51 random sentences from the NIMD corpus was manually POS-tagged, syntactically parsed and semantically annotated with 706 semantic relations.
- GlassBox Machine: the same 51 sentences used in *GlassBox Human* evaluation corpus was POS-tagged, syntactically parsed by our NLP tools and then manually annotated with 741 semantic relations.

	TreeBank	GlassBox Human	GlassBox Machine
Precision	52.32%	79.80%	66.91%
Recall	47.28%	50.82%	41.56%
F-Measure	49.67%	62.10%	51.28%

Table 5: Polaris performance results for the semantic relations evaluation corpus.

For the *Treebank* evaluation corpus, Polaris discovered 5245 relations. Of these, 2212 were exact matches to the human annotations. An additional 630 were partial matches, meaning that while the relation type was correct and the argument bracketing at least overlapped, there were

Number of Annotators	Topic	Precision		Coverage		F-Measure ($\beta = 1$)	
		Correctness	Correctness+ Relevance	Correctness	Correctness+ Relevance	Correctness	Correctness+ Relevance
3	Aquisitions	0.458762	0.369451	0.769430	0.723941	0.574805	0.489231
3	Banking	**0.663729**	0.523499	0.683941	0.614281	0.673683	0.565268
2	Finance	0.549391	0.509542	0.658332	0.638880	0.598948	0.566928
2	Illicit Drugs	0.538340	0.368481	0.763841	0.683410	0.631565	0.478802
3	Investment	0.568554	0.499937	0.784923	0.759301	0.659444	**0.602909**
1	Missiles	0.548730	0.486431	**0.823941**	**0.785210**	0.658747	0.600721
1	Terrorism	0.492831	0.389531	0.784852	0.759610	0.60547	0.514979
3	Weapons	0.636740	**0.527452**	0.756851	0.688930	**0.691619**	0.597473

Table 3: Performance results for 8 domain-ontologies generated from text.

Topic	Unique Semantic Relations					Unique Concepts		
	ISA	PW	CAU	Others	Total	In ISA/PW/CAU	Others	Total
Acquisitions	2861	1253	651	2291	7056	4398	3165	6852
Banking	2670	2138	564	2599	7971	5532	3496	7654
Finance	2757	2364	653	1746	7520	5620	2834	7292
Illicit Drugs	2842	1963	1282	4631	10718	5845	4134	8096
Investment	2253	2934	1052	3261	9500	4863	4291	9154
Missiles	3174	2621	772	2531	9098	6472	3156	8112
Terrorism	2921	3912	1525	4629	12978	7826	5312	10901
Weapons	2736	1583	732	1644	6695	4963	2751	7136

Table 4: Semantic Relation and concept extraction statistics for the evaluated ontologies presented in Table 3.

some extra or missing tokens in the generated arguments. The partial matches are scored using precision, recall, and f-measure on the overlapping tokens. For the *GlassBox Human* evaluation corpus, Polaris discovered 449 relations. Of these, 311 were perfect matches to the human annotations while 56 were partial matches. For the *GlassBox Machine* evaluation corpus, Polaris discovered 464 relations. Of these, 249 were perfect matches to the human annotations while 71 were partial matches. Table 5 presents Polaris's performance results for all the three evaluation corpora. The results include discounting for partial matches.

4.2. Ontology Evaluation

In this paper, we create and evaluate ontology libraries for intelligence (40 topics including NIPF topics) and financial (10 topics) domains. For each topic, we collected on average 500 documents from the web and manually verified their relevance to the corresponding topic. Using the technique explained in Section 3.1.2., we defined seeds sets containing on average 47 concepts of interest for each of our 50 intelligence and financial topics. We then use our methodology to create a domain-ontology for each topic, while keeping the manual intervention to a minimum.

Since the mid-1990s, various methodologies have been proposed to evaluate ontology generation/maintenance/reuse techniques (Sure et al., 2004). All the proposed methodologies have focused on some facet of the ontology generation problem, and depend on the type of ontology being created/maintained and the purpose of the ontology (Brank et al., 2005). Not much progress has been achieved in developing a comprehensive and global technique for evaluating the correctness and relevance of ontologies (Gangemi et al., 2006).

$$\mathbf{Pr(Correctness)} = \frac{\mathbf{N_j(correct)} + \mathbf{N_j(irrelevant)}}{\mathbf{N_j(correct)} + \mathbf{N_j(incorrect)} + \mathbf{N_j(irrelevant)}}$$

$$\mathbf{Pr}\begin{pmatrix} \boldsymbol{Correctness} \\ + \\ \boldsymbol{Relevance} \end{pmatrix} = \frac{\mathbf{N_j(correct)}}{\mathbf{N_j(correct)} + \mathbf{N_j(incorrect)} + \mathbf{N_j(irrelevant)}}$$

$$\mathbf{Cvg(Correctness)} = \frac{\mathbf{N_j(correct)} + \mathbf{N_j(irrelevant)}}{\mathbf{N_g(correct)} + \mathbf{N_g(irrelevant)} + \mathbf{N_g(added)}}$$

$$\mathbf{Cvg}\begin{pmatrix} \boldsymbol{Correctness} \\ + \\ \boldsymbol{Relevance} \end{pmatrix} = \frac{\mathbf{N_j(correct)}}{\mathbf{N_g(correct)} + \mathbf{N_g(added)}} \quad (1)$$

We evaluated the quality of Jaguar's domain-ontologies by comparing them against manual gold annotations. Following the ontology evaluation levels defined in (Brank et al., 2005), our evaluations are focused on the *Lexical, Vocabulary, or Data Layer* level and the *Other Semantic Relations* level. The ontologies and document collections were manually annotated by several human annotators. Viewing an ontology as a set of semantic relations between concepts, the annotators:

- Labeled an entry *correct* if the concepts and the semantic relation are correctly detected by the system else marked the entry as *Incorrect*
- Labeled a *correct* entry as *irrelevant* if any of the concepts or the semantic relation are irrelevant to the topic
- From the sentences *added new entries* if the concepts and the semantic relation were omitted by Jaguar

We use the manual annotations to compute precision (Pr) and coverage (Cvg) for the Jaguar generated domain-ontologies. The annotations also provide feedback on the automated concept tagging and semantic relation extraction modules. Equations in (1) capture the metrics defined

for the ontology evaluation. $N_j(.)$ gives the counts from Jaguar's output and $N_g(.)$ refers to gold annotation counts. Table 3 presents our evaluation results for 8 topics using a subset of 3 semantic relations defined in Table 2. Table 4 presents the semantic relation and concept extraction statistics for the eight ontologies being evaluated in this paper. The evaluation scores have been averaged over the results for different annotators. The first column in Table 3 identifies the number of human annotators for each topic. Jaguar obtained the best *Precision* in *Correctness* for the *Banking* topic. The *Weapons* topic obtained the best *Precision* for *Correctness+Relevance*. The *Missiles* topic obtained the best *Coverage* for both *Correctness* and *Correctness+Relevance*. The *Weapons* topic obtained the best *F-Measure* for the *Correctness* evaluation while the *Investment* topic obtained the best *F-Measure* for *Correctness+Relevance*.

5. Conclusions

Knowledge intensive applications require extensive domain-specific knowledge in addition to general-purpose knowledge bases. However, domain-specific ontology creation and maintenance is an expensive process and hence is referred to as the *knowledge acquisition bottleneck*. In this paper, we presented a generalized and improved procedure to automatically extract deep semantic information from text resources and rapidly create semantically-rich domain ontologies while keeping the manual intervention to a minimum. We also defined evaluation metrics to assess the quality of the ontologies created using our methodology. We presented evaluation results for a subset of the intelligence and financial ontology libraries, semi-automatically created using freely-available textual resources from the Web. The results show that a decent amount of knowledge can be accurately extracted while keeping the manual intervention in the process to a minimum.

6. Acknowledgement

This publication was made possible by a Defense Advanced Research Projects Agency (DARPA) grant. Its contents are solely the responsibility of the authors and do not necessarily represent the official view of DARPA. This publication is *DARPA Approved for Public Release, Distribution Unlimited*.

7. References

F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. P. Schneider, editors. 2003. *The Description Logic Handbook: Theory, Implementation and Applications*. Cambridge University Press, Cambridge, UK.

A. Badulescu and D. Moldovan. 2008. A semantic scattering model for the automatic interpretation of english genitives. *Natural Language Engineering*, 15(2):215–239.

C. F. Baker, C. J. Fillmore, and J. B. Lowe. 1998. The berkeley framenet project. In *Proceedings of COLING/ACL*, pages 86–90, Montreal, Canada, Aug 10-14.

M. Balakrishna and M. Srikanth. 2008. Automatic ontology creation from text for national intelligence priorities framework (NIPF). In *Proceedings of 3rd International Ontology for the Intelligence Community (OIC) Conference*, pages 8–12, Fairfax, VA, USA, December 3-4.

D. Bixler, D. Moldovan, and A. Fowler. 2005. Using knowledge extraction and maintenance techniques to enhance analytical performance. In *Proceedings of International Conference on Intelligence Analysis*, McLean, VA, USA, May 2-4.

J. Brank, M. Grobelnik, and D. Mladenic. 2005. A survey of ontology evaluation techniques. In *Proceedings of 8th International Multi-conference on Information Society*, pages 166–169, Ljubljana, Slovenia, October 10-17.

N. Choi, I. Y. Song, and H. Han. 2006. A survey on ontology mapping. *ACM Special Interest Group on Management of Data*, 35(3):34–41.

P. Cimiano. 2006. *Ontology Learning and Population from Text: Algorithms, Evaluation and Applications*. Springer, New York, NY, USA, 1st edition.

FBI. 2009. NIPF definition - Federal Bureau of Investigation (FBI): National Security Branch. `www.fbi.gov/hq/nsb/nsb_faq.htm#NIPF`.

A. Gal and P. Shvaiko. 2009. Advances in ontology matching. In *Advances in Web Semantics I: Ontologies, Web Services and Applied Semantic Web*, pages 176–198.

A. Gangemi, C. Catenacci, M. Ciaramita, and J. Lehmann. 2006. Modelling ontology evaluation and validation. In *Proceedings of Third European Semantic Web Symposium/Conference (ESWC)*, pages 140–154, Budva, Montenegro, June 11-14.

S. Harabagiu and D. Moldovan, 1998. *Knowledge Processing on an Extended WordNet*, chapter 17, pages 379–406. MIT Press.

Dan I. Moldovan and Roxana Girju. 2001. An interactive tool for the rapid development of knowledge bases. *International Journal on Artificial Intelligence Tools*, 10(1-2):65–86.

D. Moldovan, M. Srikanth, and A. Badulescu. 2007. Synergist: Topic and user knowledge bases from textual sources for collaborative intelligence analysis. In *CASE PI Conference*.

H. Pinto and J. Martins. 2004. Ontologies: How can they be built? *Knowledge and Information Systems*, 6(4):441–464.

E. Ratsch, J. Schultz, J. Saric, P. C. Lavin, U. Wittig, U. Reyle, and I. Rojas. 2003. Developing a protein-interactions ontology. *Comparative and Functional Genomics*, 4(1):85–89.

J.G. Schmolze and T. Lipkis. 1983. Classification in the kl-one knowledge representation system. In *Proceedings of 8th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 330–332, Karlsruhe, Germany.

Y. Sure, G. A. Perez, W. Daelemans, M. L. Reinberger, N. Guarino, and N. F. Noy. 2004. Why evaluate ontology technologies? because it works! *IEEE Intelligent Systems*, 19(4):74–81.

W. A. Woods. 1991. Understanding subsumption and taxonomy: A framework for progress, principles of semantic networks: Explorations in the representation of knowledge. In *Morgan Kaufmann*, pages 45–94, San Mateo, California.