Automatic Building of Semantically Rich Domain Models from Unstructured Data

Mithun Balakrishna and **Dan Moldovan**
Lymba Corporation
Richardson TX 75080 USA

Abstract

The availability of massive amounts of raw domain data has created an urgent need for sophisticated AI systems with capabilities to find complex and useful information in big-data repositories in real-time. Such systems should have capabilities to process and extract significant information from natural language documents, search and answer complex questions, make sophisticated predictions about future events, and generally interact with users in much more powerful and intuitive ways. To be effective, these systems need a significant amount of domain-specific knowledge in addition to the general-domain knowledge. Ontologies/Knowledge-Bases represent knowledge about domains of interest and serve as the backbone for semantic technologies and applications. However, creating such domain models is time consuming, error prone, and the end product is difficult to maintain. In this paper, we present a novel methodology to automatically build semantically rich knowledge models for specific domains using domain-relevant unstructured data from resources such as web articles, manuals, e-books, blogs, etc. We also present evaluation results for our automatic ontology/knowledge-base generation methodology using freely-available textual resources from the World Wide Web.

1 Introduction

The availability of massive amounts of information is transforming all aspects of human endeavor, including military, intelligence, and commercial. Users are faced with a complex task of sifting through a constant barrage of raw data to find complex and useful information in real-time. With the significant progress in the past decade in Natural Language Processing (NLP) and knowledge-based AI systems, we seem poised to enter a new age of sophisticated software systems. Such systems should have capabilities to process and extract significant information from natural language documents, search and answer complex questions, make sophisticated predictions about future events, and generally interact with users in much more powerful and intuitive ways. The dilemma we face is that for systems to understand texts and produce actionable knowledge, they need to have a significant amount of domain knowledge to start with.

Ontologies and knowledge-bases are the natural choice for encoding unstructured knowledge into accessible and actionable structured domain models that can be more easily integrated into a reasoning system (Cimiano 2006; Moldovan, Srikanth, and Badulescu 2007). While ontologies/knowledge-bases are the backbone of a number of information exchange systems, in a constantly changing world of indicators and events, creating and maintaining such domain models is a significant problem (Ratsch et al. 2003; Pinto and Martins 2004). Creation and maintenance of domain models requires significant and timely human involvement and is an error prone process (Cimiano 2006; Balakrishna and Srikanth 2008). Existing ontology/knowledge-base engineering tools require Subject Matter Experts (SMEs) to define domain concepts, events and their relations; to monitor new information and manually update the models to reflect changes to the meaning associated with the domain elements over time. This quickly becomes infeasible with the rapidly changing and vast amount of information available for a domain. Lack of automation in building and maintaining ontologies/knowledge-bases has resulted in fewer, up-to-date domain models. This is a critical bottleneck for any real-world application leveraging semantic technologies.

Existing machine-readable, open-domain dictionaries like WordNet (Miller 1995; Fellbaum 1998) lack domain specific concepts such as those that are often expressed as complex nominals, acronyms, domain specific Named Entities (NEs) and occasional slang terms. There have been previous efforts to build domain models semi-automatically or automatically using structured and unstructured data. (Gasevic et al. 2004; Bohring and Auer 2005) developed methodologies to convert structured data such as XML/UML into ontologies while (Balakrishna and Srikanth 2008; Yang and Callan 2008) built specialized ontology generation tools that are tailor-made for specific domains such as Intelligence or resources such as emails. (Hu and Liu 2004; Kong, Hwang, and Kim 2006) focused on building ontologies using a small set of semantic relations like IS-A/subClassOf, Part-Whole/Meronymy, and Synonymy. (Maedche et al. 2002; Cimiano and Volker 2005) created shallow knowledge-bases by extracting NEs and relations between these NEs.

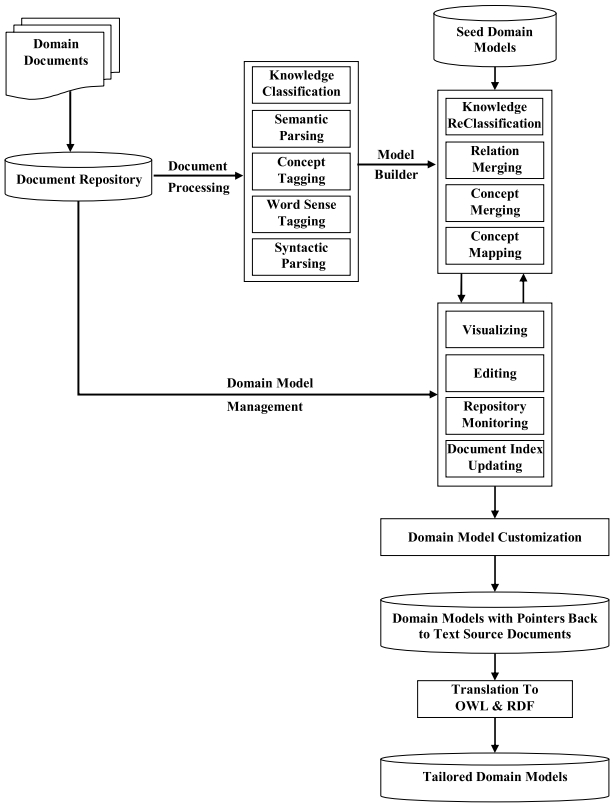

Figure 1: A framework for domain model generation, management, and customization.

Figure 1 illustrates our framework for creation, management, and customization of semantically rich domain models. The input to the system is a set of documents available in the domain repository, and one or more domain models to guide (or seed) the automatic domain model extraction process. The system starts document processing by extracting syntactic structure, word senses, key concepts, and semantic relations from new or updated documents. This information is organized into a hierarchy and semantic contexts using knowledge classification algorithms. The domain model builder consults one or more existing domain models, and for each processed document performs concept mapping, relation and concept merging, and knowledge classification against each domain model. The resulting models contain new or updated information based on the content of the processed documents with pointers back to the document source for each concept and relation. The new information is translated into RDF and OWL for semantic indexing and/or review. Domain model management is required when a user wants to visualize and edit a model, and when new document concepts or relations are added to a model. Model customization is important for users that want to create their own set of relations on top of those that are part of the domain models without having to re-process all the documents and train a new semantic parser. This allows each application to define its own view of the data for a tailored analysis without requiring a separate repository.

In this paper, we present a novel methodology to automatically build semantically rich domain models from unstructured data resources such as web articles, manuals, e-books, blogs, etc. We concentrate and present in detail the following two modules in our automatic domain model creation framework illustrated in Figure 1: 1) Domain relevant concept and semantic relations extraction from text; 2) Hierarchy creation using knowledge classification algorithms.

2 Domain Relevant Concept and Semantic Relations Extraction from Text

The first step in building any domain model is to identify key and relevant domain concepts and the relationships between these domain concepts. We extract rich semantic information from the text of a domain-relevant document collection using a starter/seed domain model containing some sample domain-relevant concepts and semantic relations.

Relation	Definition
Agent(X,Y)	X is the agent of Y; X is prototypically a person
Association(X,Y)	Person X is associated with Person Y; the relation is not necessarily kinship
Cause(X,Y)	X causes Y
Experiencer(X,Y)	X is an experiencer of Y; involves cognition and senses
Influence(X,Y)	X caused something to happen to Y
Instrument(X,Y)	X is an instrument in Y
Intent(X,Y)	X is the intent/goal/reason of Y
IS-A(X,Y)	X is a (kind of) Y
Justification(X,Y)	X is the reason or motivation or justification for Y
Kinship(X,Y)	X is a kin of Y; X is related to Y by blood or by marriage
Location(X,Y)	X is location of Y or where Y takes place
Make(X,Y)	X makes Y
Manner(X,Y)	X is the manner in which Y happens
Part-Whole(X,Y)	X is a part of Y
Possession(X,Y)	X is a possession of Y; Y owns/has X
Property(X,Y)	X is a property/attribute/value of Y
Purpose(X,Y)	X is the purpose for Y
Quantification(X,Y)	X is a quantification of Y; Y can be an entity or event
Recipient(X,Y)	X is the recipient of Y; X is an animated entity
Source(X,Y)	X is the source, origin or previous location of Y
Stimulus(X,Y)	X is the stimulus of Y; Perceived through senses
Synonymy(X,Y)	X is a synonym/name/equal for/to Y
Theme(X,Y)	X is the theme of Y
Time(X,Y)	X is time of Y or when Y takes place
Topic(X,Y)	X is the topic/focus of cognitive communication Y
Value(X,Y)	X is the value of Y

Table 1: The set of 26 semantic relations used in our automatically generated domain models.

The domain documents are processed through the following deep semantic NLP tools pipeline: word boundary detection, part-of-speech tagging, sentence boundary detection, named-entity recognition, chunk syntactic parsing, full syntactic parsing, word-sense disambiguation, co-reference resolution, semantic parsing, and event extraction. Table 1 lists the set of 26 semantic relation types extracted by the Semantic Parser for the purposes of text understanding. Se-

mantic relations are abstractions of underlying relations between concepts, and provide connectivity between concepts and contexts (Badulescu and Moldovan 2009). The 26 relations cover most of the thematic roles proposed by Fillmore and others, and the semantic roles in PropBank. To find semantic relations in text, the parser uses an hybrid approach combining state-of-the-art in text processing, pattern matching and machine learning techniques (Badulescu and Moldovan 2009; Balakrishna et al. 2010). Together, the set of 26 semantic relation types can give a structured picture of the specified event: who was involved, what was done, and to whom; and for what purpose.

As the input documents are processed through the NLP tools, concepts that are part of the original domain model will guide the search space for new concepts and semantic relations. We use these seed concepts to examine all the concepts and semantic relations extracted by NLP tools from the input document set and filter them for relevance to the domain. We compute a domain model relevance score for each discovered concept based on the semantic distance from the original seed relations and concepts. The semantic distance is a function of the number and type of relations and concepts that are on the semantic paths that link the original domain model concepts to the newly discovered candidate concepts. Only concepts with a semantic distance score above a user selected threshold will be added to the domain model. In order to accommodate information added or deleted from the document set, we track changes in the repository and update the information in the domain model.

The following steps present our algorithm to automatically extract domain relevant concepts and semantic relations from text:

1. Process the input domain-specific document collection to extract text from the documents and then filter/clean-up the extracted text. The input includes a variety of document types(e.g. 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.

2. Process the input domain documents through the previously mentioned NLP tools pipeline.

3. In an NLP processed document, identify sentences containing seed concepts.

4. In the sentences selected from Step 3, identify noun phrases that contain the seed concept word(s) and phrase(s) semantically linked to another noun/verb seed concept word(s) by any semantic relations (e.g., IS-A, AGENT, PART-WHOLE, SYNONYMY) as key concept candidates.

5. Every noun phrase identified in Step 4 is considered to be a potential new concept. Noun phrases are then processed to extract well-formed noun concepts using syntactic pattern rules.

5.1 Collocations: search the noun phrase for word collocations that are defined in WordNet as a concept. E.g. *nuclear_weapon*, *hand_grenade*, etc. can be extracted as well-formed concepts. If a concept is present in WordNet then a normalized form of the concept is used to represent the concept (and its synset concepts) in the domain model. E.g. *weapon of mass destruction* will represent all occurrences of *weapons of mass destruction* or *WMD* or *WMDs* or *W.M.D.*

5.2 Named Entities: search the noun phrases for named-entities and extract them as concepts while preserving the case from the text or converting the concept into Title Case if the text is in lower case in the document. E.g. *george bush* is extracted as a *human* and normalized to *George Bush*.

5.3 Descriptive Adjective Filtering: when adjectives are part of the noun phrases, extract as concepts only those noun phrases that are formed with relational and participial adjectives while the noun phrases with descriptive adjectives are discarded since descriptive adjectives do not add important information to the nouns that they modify. Concepts like *british tea* (relational adjective based) and *boiling water* (participial adjective based) are extracted while concepts like *fast growth* and *high interest* are discarded.

5.4 Verbal Modifier based Extraction: verbs in certain tense forms can modify nouns resulting in creation of complex nominals. For example, past participles tagged as VBD part-of-speech: *bombed city*, *robbed bank*, etc; and present continuous tagged as VBG part-of-speech: *bombing crew*, *robbing gang*, etc. Some of these concepts are also marked as adjectives in a resource (e.g. WordNet) and will be included in the default concept extraction but since these modifiers are marked as verbs, we require special patterns to enable their extraction.

5.5 Plural Normalization: plurals are normalized to their singular counterparts using WordNet. For concepts that do not occur in WordNet, we split the concepts based on space, hyphen, and other punctuations, and perform a WordNet check on the postfix word combinations.

5.6 Determiner and Numeral Filtering: search the noun phrase and prevent the determiner/numeral nodes from being part of any concept under that noun phrase.

5.7 Concept Splitting: if a conjunction or some concept-delimiting punctuation like "," or ":" is found under a noun phrase, split the noun phrase to create two concepts at the point of the conjunction or punctuation.

6. In the sentences selected from Step 3, identify verb phrases that contain the seed concept word(s) and phrase(s) semantically linked to another noun/verb seed concept word(s) by any semantic relations (e.g., IS-A, AGENT, PART-WHOLE, SYNONYMY) as key concept candidates.

7. Augment the seed words with Step 5 and 6's domain concepts and return to Step 3. The process of sentence selection, concept extraction, semantic relation extraction, and seed concepts set augmentation is repeated iteratively *n* number of times (by default, n = 3).

8. Collect all relations that link the identified domain concepts with other concepts (in- or out-of-the-domain). Relations between domain concepts become part of the model.

9. Classify each concept occurrence/mention as *Class* or *Instance* using linguistic and contextual clues e.g., plurals and no-modifier mentions are indicative of class references. In addition to lexical clues, named entities, word senses, semantic relations, and event attributes are used as machine learning features into an SVM classifier to label a concept.

3 Knowledge Classification Algorithms

In this section, we describe our automatic concept hierarchy creation process that organizes and connects the domain relevant concepts and semantic relations extracted in Section 2. We present classification procedures to classify concepts against each other in order to generate a hierarchy connecting concepts. Classification determines where in an concept hierarchy a given concept fits. The domain model contains concepts linked not just with subClassOf/IS-A relations but contains a rich semantic network connecting the property-rich concepts through other semantic relations including PART-WHOLE, CAUSE and SYNONYMY. Our methodology forms a hierarchy using the extracted concept and relation information via transitive semantic relations that generally hold to be universally true. Example: Given the concepts *weapon*, *nuclear weapon*, *hydrogen bomb*, and *thermonuclear weapon*, and semantic relations *nuclear weapon* **IS-A** *weapon* and *hydrogen bomb* **IS-A** *nuclear weapon*, our algorithm forms the following hierarchy: *hydrogen bomb* $\overrightarrow{ISA}$ *thermonuclear weapon*. $\overrightarrow{ISA}$ *nuclear weapon* $\overrightarrow{ISA}$ *weapon*. The classification is based on the subsumption principle (Schmolze and Lipkis 1983; Woods 1991; Baader et al. 2003) and Textual Entailment (Tatu and Moldovan 2005; Tatu et al. 2006).

From the discovered set of semantic relations, our classification methodology considers all the IS-A and SYNONYMY relations. There are two distinct possibilities:

1. A IS-A or SYNONYMY 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.

2. 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.

Using the hierarchy forest obtained from the above steps, we run several knowledge classification procedures on concepts that do not link to WordNet directly or indirectly. Our classification methods that derive SYNONYMY (SYN) and IS-A relations exploit the compositional meaning of non-WordNet domain concepts.

SYNONYMY and IS-A Derivation Procedure

1. For any two domain concepts of the form *modifier head* and *head*, we create a IS-A($modifierhead, head$) relation. We consider only those head nouns and adjectives that do not have any hyponyms (Miller 1995). More complex cases such as 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 *modifier head* is ontologically subsumed by concept *head* e.g. the concept *nontaxable dividends* is subsumed by *dividends*.

2. For any two domain concepts, c_i, of the form $modifier_i$ $head_i$ (i=1,2):

2.1 If IS-A($modifier_1, modifier_2$) and IS-A ($head_1, head_2$), then IS-A(c_1, c_2). e.g. *Japan discount rate* is subsumed by *Asian country interest rate*.

2.2 If IS-A($modifier_1, modifier_2$) and SYN ($head_1, head_2$), then IS-A(c_1, c_2). e.g. *IS-A (Japan discount rate, Asian country discount rate)*.

2.3 If SYN($modifier_1, modifier_2$) and IS-A($head_1, head_2$), then IS-A(c_1, c_2). e.g. *Japan discount rate* is subsumed by *Japan interest rate*.

2.4 If SYN($modifier_1, modifier_2$) and SYN ($head_1, head_2$), then SYN(c_1, c_2).

The IS-A or SYNONYMY relation links between ($modifier_1, modifier_2$) and ($head_1, head_2$) may not always be a direct and may consist of a chain of IS-A or SYNONYMY relations since they are transitive relations. If there is no direct relation in WordNet between ($modifier_1, modifier_2$) and ($head_1, head_2$), but there are common subsuming concepts. Then, we pick the Most Specific Common Subsumer (MSCS) concepts of ($modifier_1, modifier_2$) and ($head_1, head_2$), respectively. Then form a concept [$MSCS(modifier_1, modifier_2), MSCS(head_1, head_2)$] and place $modifier_1$ $head_1$ and $modifier_2$ $head_2$ under it. e.g. to classify *Japan discount rate* with respect to *Germany prime interest rate*, we add the concept *country interest rate* to the hierarchy and place both the concepts *Japan discount rate* and *Germany prime interest rate* under it.

3. To classify a concept $modifier_1$ $modifier_2 head$:

3.1 If there is already a concept $modifier_2$ $head$ in the knowledge base under the concept $head$, then we will place $modifier_1$ $modifier_2$ $head$ under concept $modifier_2$ $head$. E.g. to classify the concept *nuclear weapons testing*, if we have the concept *weapons testing* under *testing* then we add the concept *nuclear weapons testing* under *weapons testing*.

3.2 If there is already a concept $modifier_1$ $head$ in the knowledge base under the concept $head$, then place $modifier_1$ $modifier_2$ $head$ under concept $modifier_1$ $head$. E.g. to classify the concept *nuclear weapons testing*, if we have the concept *nuclear testing* under *testing* then we add the concept *nuclear weapons testing* under *nuclear testing*.

3.3 If both the above cases are $true$ then place $modifier_1$ $modifier_2$ $head$ under both concepts $modifier_2$ $head$ and $modifier_1$ $head$. E.g. to classify *nuclear weapons testing*, if we have the concept *nuclear testing* under *testing* and *weapons testing* under *testing* then we add the concept *nuclear weapons testing* both under *nuclear testing* and under *weapons testing*.

4. Classify a concept $modifier_1$ $head$ with respect to a hierarchy under the concept $head$. The task is to identify the Most Specific Subsumer (MSS) from all the concepts under the concept $head$ that subsumes $modifier_1$ $head$. By default, $modifier_1$ $head$ is placed under $head$, however, since it may be more specific than other hyponyms of $head$, a more complex classification analysis is needed. 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

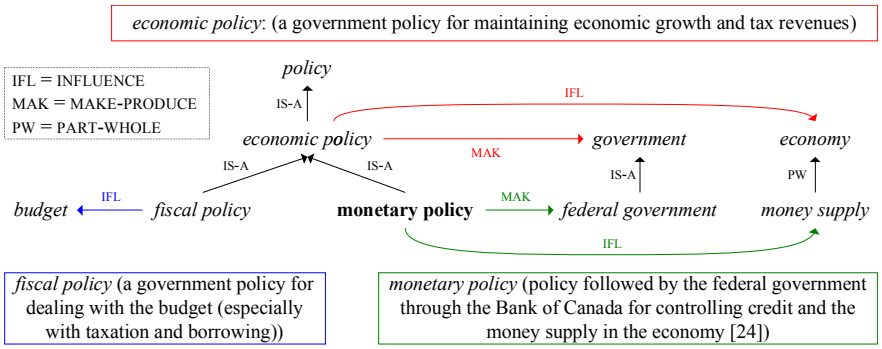

Figure 2: Classification of a new concept *monetary policy* using our textual entailment based concept subsumption method.

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 the concepts A and B:

4.1 Extract verb relations between concept and other gloss concepts. E.g. $AR_{a1}C_{a1}$, $AR_{a2}C_{a2}$,....,$AR_{am}C_{am}$; $BR_{b1}C_{b1}$, $BR_{b2}C_{b2}$,....,$BR_{bn}C_{bn}$

4.2 A subsumes B if and only if:

4.2.1 Relations R_ai subsume R_bi, for $1 \leq i \leq m$

4.2.2 C_{ai} subsumes or is a meronym of C_{bi}

4.2.3 Concept B has more relations than concept A, i.e. $m \leq n$

Concept Subsumption Based on Comparisons of Semantic Models

In addition to the classification methods described above, we use concept classification algorithms that compare the contextual semantic models of two domain concepts - these sets of semantic relations partly define the concepts. For two domain concepts A and B and their contextual models, $R_{A1}(A, C_{A1})$, $R_{A2}(A, C_{A2}), ...$, $R_{An}(A, C_{An})$ and $R_{B1}(B, C_{B1})$, $R_{B2}(B, C_{B2}), ...$, $R_{Bm}(B, C_{Bm})$, we will determine the degree of subsumption between A and B based on the type of semantic relations R_{Ai} $(i \in 1, ..., n)$ for which there exists an index $j \in 1, ..., m$ such that IS-A(C_{Ai}, C_{Bj}) and $R_{Ai} = R_{Bj}$. For high subsumption degrees, we have IS-A*(B,A)*. Similarly, a synonymy degree can be computed. We highlight the dependency of the subsumption/synonymy degree on the type and proportion of matched relations (R_{Ai}). Concepts that share SYNONYMY relations are candidates for a SYNONYMY relation.

Textual Entailment for Concept Subsumption

For domain concepts which have a textual definition or gloss, we include novel classification mechanisms that use textual reasoning to determine whether the definition of a concept semantically entails the description of another concept and create an IS-A relation when required. Symmetric entailment will generate a SYNONYMY relation. Let us consider the classification of the concept *monetary policy* with respect to the hierarchy *fiscal policy* IS-A *economic policy* IS-A *policy*. By default, this concept is placed under *policy*. However, if we consider each concept's definition *monetary policy* is subsumed by *economic policy* because *government* subsumes *federal government* and *money supply* is part of the *economy*. Figure 2 shows this classification of a new concept *monetary policy* using our textual entailment based concept subsumption methodology.

4 Evaluations

Since the inception of domain modeling techniques, several methodologies (Sure et al. 2004; Brank, Grobelnik, and Mladenic 2005; Gangemi et al. 2006) have been proposed to evaluate correctness and relevance. These evaluations have focused on some facet of the model generation problem depending on the model type and purpose. For our system evaluation, we picked 4 domains (2 intelligence and 2 financial domains), and randomly identified 1000 sentences for each domain from a set of 100 domain relevant documents that are freely-available on the Web. Two SMEs (for each topic) annotated their corresponding domain repository sentences with relevant concepts and semantic relations. They also manually created models for each domain by only using the identified domain sentences as a reference. The SMEs also defined seeds sets containing 20 concepts of interest for each topic. In this section, we compare the output of our concept and semantic relation extraction, and hierarchy creation against the manual gold annotations.

We evaluated the correctness and relevance of our domain concept and relation extraction (Section 2) at the *Lexical, Vocabulary, or Data Layer* and *Other Semantic Relations* levels (Brank, Grobelnik, and Mladenic 2005) using the precision and coverage metrics defined in (Balakrishna et al. 2010). SMEs validated the domain concepts and semantic relations automatically extracted from each domain sentence. Table 2 presents the results for our automatic extraction capabilities against the manual gold annotations for each sentence. Our automatic knowledge classification procedure (Section 3) created concept hierarchies for each domain using the gold concept and semantic relation annotations created by the SMEs for that corresponding domain. We then evaluated each domain hierarchy against its corresponding manual concept hierarchy, at the *Most Specific Subsumer (MSS)* level (Balakrishna et al. 2010). These evaluation results are presented in Table 2. Results show that

Domains	Concept and Semantic Relation Extraction Evaluation						Hierarchy Evaluation		
	Precision		Coverage		F-Measure		Precision	Coverage	F-Measure
	Correctness	Correctness + Relevance	Correctness	Correctness + Relevance	Correctness	Correctness + Relevance	Correctness	Correctness	Correctness
Weapons	0.692990	0.619481	0.779265	0.706893	0.7336	0.660307	0.853801	0.714752	0.778113
Banking	0.728591	0.651090	0.702471	0.659110	0.715293	0.655075	0.865167	0.730028	0.791873
Illicit Drugs	0.659125	0.594660	0.738011	0.669001	0.696341	0.629644	0.808311	0.621023	0.702397
Finance	0.625210	0.603619	0.729847	0.699461	0.673488	0.648015	0.842149	0.685190	0.755604

Table 2: Performance results for the automatic domain concept and relation extraction, and automatic hierarchy creation.

our automatic methodology can extract 68.5% (average) of domain relevant knowledge encoded in the text with 61.5% (average) accuracy, and create 68.75% (average) of the domain hierarchy with 84.25% (average) accuracy.

5 Conclusions

The availability of massive amounts of raw domain data has created an urgent need for sophisticated applications that require extensive domain-specific knowledge along with existing general-domain knowledge. In this paper, we presented a novel methodology for building semantically rich domain models from easily available unstructured data. We presented a module that automatically extracts domain relevant concept and semantic relations from text. We then presented a hierarchy creation module that uses novel knowledge classification algorithms. We evaluated the performance of our automatic domain model creation methodology on intelligence and financial domains using freely-available textual resources from the Web. Our results show that a good amount of knowledge can be accurately and automatically extracted from text into domain-specific knowledge models, and hence easing the knowledge acquisition bottleneck.

6 Acknowledgement

This research was made possible by a United States Air-Force (USAF) grant. The views and conclusions in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the USAF or the U.S. Government.

References

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

Badulescu, A., and Moldovan, D. 2009. A semantic scattering model for the automatic interpretation of english genitives. *Natural Language Engineering* 15(2):215–239.

Balakrishna, M., and Srikanth, M. 2008. Automatic ontology creation from text for national intelligence priorities framework (NIPF). In *Proceedings of OIC*, 8–12.

Balakrishna, M.; Moldovan, D.; Tatu, M.; and Olteanu, M. 2010. Semi-automatic domain ontology creation from text resources. In *Proceedings of LREC*.

Bohring, H., and Auer, S. 2005. Mapping xml to owl ontologies. In *Proceedings of Leipziger Informatik-Tage*, 147–156.

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

Cimiano, P., and Volker, J. 2005. Text2onto - a framework for ontology learning and data-driven change discovery. In *Proceedings of NLDB*, volume 3513, 227–238.

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

Fellbaum, C., ed. 1998. *WordNet: An Electronic Lexical Database*. MIT Press.

Gangemi, A.; Catenacci, C.; Ciaramita, M.; and Lehmann, J. 2006. Modelling ontology evaluation and validation. In *Proceedings of Third European Semantic Web Symposium/Conference (ESWC)*.

Gasevic, D.; Djuric, D.; Devedzic, V.; and Damjanovic, V. 2004. From uml to ready-to-use owl ontologies. In *2nd International IEEE Conference In Intelligent Systems*, volume 2, 485–490.

Hu, H., and Liu, D.-Y. 2004. Learning owl ontologies from free texts. In *Proceedings of International Conference on Machine Learning and Cybernetics*, volume 2, 1233–1237.

Kong, H.; Hwang, M.; and Kim, P. 2006. Design of the automatic ontology building system about the specific domain knowledge. In *Proceedings of 8th ICACT Conference*, volume 2.

Maedche, A.; Motik, B.; Silva, N.; and Volz, R. 2002. Mafra - mapping distributed ontologies in the semantic web. In *13th European Conf. Knowledge Eng. and Management (EKAW 2002)*, 235–250.

Miller, G. 1995. Wordnet: a lexical database for english. *Communications of the ACM* 38(11):39–41.

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

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

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

Schmolze, J., and Lipkis, T. 1983. Classification in the kl-one knowledge representation system. In *Proceedings of IJCAI*.

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

Tatu, M., and Moldovan, D. 2005. A semantic approach to recognizing textual entailment. In *Proceedings of HLT-EMNLP*.

Tatu, M.; Iles, B.; Slavick, J.; Novischi, A.; and Moldovan, D. 2006. Cogex at the second recognizing textual entailment challenge. In *Second PASCAL Challenges Workshop on RTE*.

Woods, W. A. 1991. Understanding subsumption and taxonomy: A framework for progress, principles of semantic networks: Explorations in the representation of knowledge. In *Morgan Kaufmann*.

Yang, H., and Callan, J. 2008. Ontology generation for large email collections. In *Proceedings of International Conference on Digital Government Research*, 254–261.