Building Automatically a Business Registration Ontology

Melania Degeratu
Department of Computer Science
Columbia University
New York, NY 10027, USA
melania@cs.columbia.edu

Vasileios Hatzivassiloglou
Department of Computer Science
Columbia University
New York, NY 10027, USA
vh@cs.columbia.edu

Abstract

We discuss a domain-independent, corpus based method for dictionary-less automatic extraction of ontological knowledge from domain-specific unannotated documents. We present the architecture, algorithms, and results for ONTOSTRUCT—a new system that uses machine learning and statistical techniques to analyze text sources, discover terms, link equivalent terms into concepts, learn both hierarchical and non-hierarchical conceptual relations, and build an extensive, semantically sound hierarchy of concepts. We report on ONTOSTRUCT's results in constructing a domain-specific ontology for the business registration domain, and evaluate the performance of two of its modules.

1 Introduction

The New Jersey State government has adopted an initiative in electronic commerce, with the goal of providing an effective and efficient electronic interface that fosters the establishment of new businesses and facilitates effective long-term interactive relationships with businesses throughout the State [Adams *et al.* 2000]. The state government envisions the opening of a web portal that will manage all interactions between businesses and various New Jersey State agencies. The services offered by such a web portal will range from the establishment of a business to maintenance and reporting on a periodic basis. Given the large number of state agencies providing relevant information sensitive to the registration and establishment of a business entity, it is extremely important to provide a common foundation of terms and mappings between related terms used by different state agencies, i.e., an ontology for that domain. The encapsulation of dependency and specialization relations captured in the ontology allows the automatic adaptation of general interface schemas (e.g., presenting to the user all the items needed for registration of a business, and only valid choices for them).

Historically, ontologies have been manually generated, especially as it pertains to the discovery of equivalent terms and relations. The manual construction of a large-scale ontology such as WordNet [Miller *et al.* 1990] or CYC [Lenat 1995] involves significant human effort and requires access to domain experts and specialist knowledge engineers. Moreover, this process is hard to scale up, and it is not portable across different domains. These problems have prompted researchers to develop semi-automated processes (for example, for the enrichment of SENSUS [Knight and Luk 1994]) and even explore fully automatic methods for some ontology building subtasks (e.g., [Sanderson and Croft 1999; Agichtein and Gravano 2000; Bisson *et al.* 2000]). The resources used include existing ontologies, free text sources, online thesauri and dictionaries, and databases representing domain relationships.

In this paper we present ONTOSTRUCT—a system using domain-independent, corpus based methods for the automatic extraction of ontological knowledge from domain-specific unannotated documents. Given the richness of information encoded in unstructured natural language text, we explore methods for recovering terminological and semantic information about a given domain automatically from these text documents only. ONTOSTRUCT does not rely on any pre-existing knowledge resource such as dictionaries, term lists,

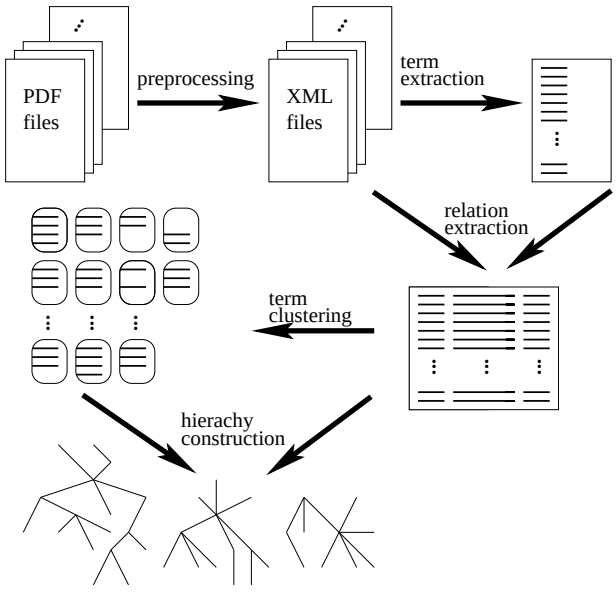

Figure 1: ONTOSTRUCT architecture.

database schemas, or semantic models of the domain. It performs all three core component tasks for ontology construction (identifying terms, linking them into concepts, and learning relations between terms and concepts), and does so in a way that allows its immediate application to a different domain once data from that domain becomes available. We present the data and algorithms used by our system in Section 2, and report on its results and performance on a preliminary evaluation in Section 3. Our results indicate that ONTOSTRUCT achieves precision over 70% in relationship extraction, while producing groups of linked concepts of which almost half have no inappropriate members.

2 System Description

The ONTOSTRUCT architecture, shown in Figure 1, contains the following modules: text preprocessing, term extraction, relation extraction, information clustering, and ontology management.

2.1 Data

The resources available to us are primarily texts that the agencies already use for supporting current business registration and regulation policies. These documents are not unlimited in number or especially large in size, so one direction that our research departs from other efforts is that we looked for techniques that would work with such data. Unlike mining the web for generally reported information such as company-address relations [Agichtein and Gravano 2000] or general semantic relationships of English words [Knight and Luk 1994], we have a substantially more limited and non-extensible collection of documents. On the other hand, this textual data tend to have additional structure (e.g., lists of items, question and answer pairs, instructions for filling in blanks) that is not as prevalent in most text resources, and we have designed ONTOSTRUCT to take advantage of that partial structure. We have used for all experiments reported in this paper pre-existing text documents provided by the state agencies, mainly business registration forms, permit application forms, and taxation forms, and their corresponding filling instructions. We have compiled a corpus of 57 PDF (Portable Document Format) documents with a total of 115,865 words. Since the documents were developed by independent state agencies with no common structural and formatting guidelines, and they were

intended for manual handling by both users and state employees, there is a great variety in their content, appearance, style, and structure, which adds to the difficulty of the problem. As a consequence, we cannot develop a generalized method of extracting information based on the document's layout, and have to rely on weaker, local techniques for capturing structural information (e.g., by detecting itemized lists). Another important point arising from the analysis of the this corpus, is the lack of multiple documents dealing with the same topic. While the documents are related, each of them focuses on information about a specific form, information that may or may not be generalizable to the other forms.

2.2 Information Extraction

Preprocessing We first have to translate the input PDF documents to ASCII text, which is achieved by automatically submitting the PDF documents to the online conversion tool available at the Adobe Acrobat Accessibility web page (`http://access.adobe.com/onlinetools.html`). We use a locally trained model of *Mxterminator* [Reynar and Ratnaparkhi 1997] to detect sentence boundaries, and then apply a bootstrapping algorithm that looks for increasing sequences of seed "counter" words, and learns leading words which function as list markers. We identify correct occurrences of 1,642 list markers, such as "question", "item", and "table", and mark them in the text. The extracted list markers provide valuable information towards topic detection, as we can consider them to divide the text in topic-centered paragraphs. Since these paragraphs can also be nested, we can extract a tree-like structure containing locality information. We say that two sentences are locally related if they appear on the same subtree in the tree; their locality coefficient is given by the number of shared nodes in the paths from the root to each of the sentences. This locality information will be used later when we compute the dissimilarity measure used for term clustering.

We then use a part-of-speech tagger and chunker [Finch and Mikheev 1997] to assign tags to content and function words, and to identify noun and verb phrases in the sentences. For each word in the text we also use automatic tools (our local implementation of the Porter algorithm [Porter 1980]) to obtain the stem—a string common to morphologically related words. We also record the derivational root for all the verbs in the text. The purpose of these steps is to conflate all linguistic forms of a term in order to facilitate matching. To ease this process of shallow document parsing, all information collected so far is automatically encoded in XML format. These XML documents are the input data for the subsequent modules of ONTOSTRUCT.

Term Extraction During the term extraction stage, the system extracts candidate terms from the pool of noun phrases identified earlier by the chunker. For each such noun phrase we use grammatical filters to compute a standardized form, which contains only the nouns and their adjectival modifiers. Given the domain specificity, we select not to eliminate adjectives from the standardized form of the noun phrases. The motivation behind this choice is that for many cases in the business registration domain, the adjective conveys a significant amount of information. For example, we learn that different provisions apply for "domestic profit" than for "foreign profit". We compute the frequency of all standardized noun phrases, and we select as terms all those noun phrases that appear at least twice in one of the documents, or appear in at least two different documents. Such distributional characteristics have been used successfully before for term identification [Justeson and Katz 1995].

A separate set of terms, which do not have to obey the aforementioned frequency constraints, is the set of noun phrases acting as objects of imperative verbs. The most frequent imperative verbs occurring in this corpus are "enter" and "indicate", whose high rate of occurrence is easily explained as the documents include instructions for filling forms. The two term extraction methods produce a combined total of 9,356 distinct extracted terms from the 33,005 noun phrases found in our corpus.

Relationship Extraction The ontological relations we want to extract are both hierarchical and non-hierarchical. In order to derive conceptual relations between the terms extracted from our corpus, we first

Leading Term	Enumeration
new business entity	corporation, limited liability company, limited partnership, limited liability partnership
officer	president, VP, chief executive officer
information services	news syndicates, libraries, on-line information services

Figure 2: Enumeration patterns inducing hierarchical relations.

Term	Attributes
employer	federal employer identification number
employer	New Jersey taxpayer identification number
employee	social security number
employee	gross wages

Figure 3: Attributes extracted by the algorithm

analyze various predefined syntactic patterns occurring in the text. An example of a predefined pattern is the enumeration of terms, which can be easily recognized using regular expressions. We observe, for example, that 71% of enumerations with at least three terms appear in a parenthetical pattern following a noun phrase. These enumerated terms are likely to be specialized concepts of the noun phrase before the start of the enumeration, i.e., representing *is-a* relationships. Examples of such patterns are shown in Figure 2.

Using parenthetical patterns, we can also extract equivalence relations between a term and its acronym. The acronym of a multi-word term is easily obtained in many cases by putting together the first letter from each term word. Examples of acronyms extracted with this method are FEIN for "federal employer identification number" and SIC for "standard industrial code".

Term attributes can also be extracted using contextual information. Attributes appear in text in one of the following forms: <*attr*> *of* <*term*> (e.g., "name of person") or <*term*>*'s* <*attr*> ("person's name"). We find occurrences of these patterns from the text, and we create lists of attributes for all the terms in the corpus. An example of such a list is presented in Figure 3.

The most frequent relations in free text are characterized by predicative links. Terms are predicatively linked to a verb to which they serve as subject or object. We extract relations given by the following types of patterns: <*term*> <*verb*>, <*verb*> <*term*>, and <*verb*> <*preposition*> <*term*>. For each term in the corpus we then compile a list of verbs for which the term plays the role of the subject, and a list of verbs for which the term is the object of the verb. Examples are presented in Figure 4.

2.3 Ontology Construction

Different terms that are equivalent or nearly equivalent in meaning are often used by different data sources, in our case different state agencies. ONTOSTRUCT contains a module for detecting term equivalence and forming conceptual classes on the basis of the similarity of the relationships in which extracted terms participate. We use the complete link hierarchical clustering algorithm, with a variant of the Lance and Williams' coefficient [Lance and Williams 1967] as a dissimilarity measure. The objects being clustered are the terms, while the attributes and the verbs co-occurring with the terms are used to compute the dissimilarity measure between terms. This dissimilarity measure is computed as an equally weighted sum of the dissimilarities of each of the three lists of attributes, verbs where terms are subjects, and verbs where the terms are objects. It is then converted to a similarity value by subtracting from one, and the similarity is further adjusted by

Subject	Verb	Object
person	file	claim
person	file	contribution report
person	file	new business certificate
corporation	file	form CBT-2553

Figure 4: Verbal relations extracted by the algorithm

municipal clerk, county clerk, newark mayor
building, bureau, division
refund, tax, contribution
boy, girl, infant, man, woman
manager, applicator, owner, manufacturer, worker

Figure 5: Term clusters extracted by the algorithm

multiplying it with the locality coefficient of the two terms (as defined in Section 2.2). The cut value for the hierarchical clustering algorithm is automatically set at the point when the number of clusters is halved, or at the point when the size of the largest cluster is bigger than a threshold, whichever comes first. To eliminate overgeneralization, we set this threshold value to be 6. Examples of clusters obtained using this clustering algorithm are shown in Figure 5.

The classes of more than two terms obtained as the result of the clustering algorithm are then considered as part of the input for the hierarchy construction algorithm. The other input is the class of hierarchical relations learned in the relation extraction module. We filter these ordered pairs of hierarchically related terms such that we only retain those pairs for which at least one component is part of one term class, or the pairs that have at least one component in common with at least another class. The result of the algorithm will be a forest of hierarchies that form a directed acyclic graph. The algorithm starts with a list of nodes represented by the classes of terms and for each unused ordered pair we search for its articulation point and we add the corresponding edge to the graph. The filtering stage assures us that for each pair there is at most one articulation point. If the resulting graph doesn't satisfy the acyclicity condition, we merge all the terms encountered by traversing the newly formed cycle and converting it into a new node.

The produced trees are then enriched by adding attributes to each node. Since hierarchical relations propagate attributes through inheritance down to the leaves, the number of attributes for the nodes in the resulting forest of trees will increase with the tree depth.

3 Results and Evaluation

We have applied the five modules of the ONTOSTRUCT system to our corpus of 57 PDF documents from the business registration domain. The system extracted 9,356 terms out of 33,005 detected noun phrases. It found 123 hierarchical links, 759 attributes, and 3,106 other conceptual relations, for a total of 3,981 relations. 3,681 of the 9,356 terms (39%) were involved in one or more of these relations. In addition, 293 terms were grouped into 127 conceptual classes; classes had between 2 and 6 members, with an average of 2.31. Note that in comparison to other text mining system, the ratio of extracted information (terms and relations) to the number of words in the corpus (115,865) is relatively high; this is a consequence of both the semi-structured information present in the forms and our design decisions to extract information with relatively few instances of strong evidence.

Evaluating the performance of ONTOSTRUCT is a daunting task, because there is no pre-existing stan-

dard for what the correct output should be in this specialized domain, in the form of an ontology or otherwise. Unlike systems that aim to produce general knowledge that can be compared against existing data sets, ONTOSTRUCT's output must be evaluated on a case by case basis, preferably by domain experts. We have performed two small experiments to measure the quality of ONTOSTRUCT's results for two of the most crucial modules, and we are in the process of setting up a more formal evaluation involving people from the New Jersey state agencies. For the current preliminary experiments, we evaluate only precision (as recall is very hard to estimate without a reference standard—the evaluators would have to attempt to recover the information themselves from the textual data), and rely on judgments from two non-experts with fluent knowledge of the English language but not prior expertise of the domain.

We have randomly selected a test set of 200 relations out of the 3,981 relations produced by the algorithm, such that the relation type distribution in the test set follows the general distribution of types in the entire data set. Each relation was evaluated by two human judges, and the agreement between the human judges was measured using κ—the kappa coefficient of agreement [Siegel and Castellan 1988]. κ ranges between -1 and $+1$, and it measures the rate of agreement corrected for chance agreement. For our corpus of 200 relations, we have obtained $\kappa = 0.71$, which indicates significant agreement. The first judge considered 146 relations out of the total of 200 (73%) to be correct, while the second judge found 151 relations to be correct, for a precision of 75%.

In the next step of the evaluation process judges were asked to assess the quality of the term classes produced by the clustering algorithm. The system has produced 127 clusters with a mean size of 2.31 terms per cluster. From these, 50 clusters were randomly selected as our test set. The judges were instructed to consider a cluster to be correct only if all the terms present in the cluster are strongly semantically related.

The coefficient of agreement for the two judges performing this task was $\kappa = 0.46$, indicating weaker agreement but still clearly above chance. The precision as measured by the first judge is 36%, corresponding to 18 correct clusters out of the presented 50. The second judge identified 21 clusters to be correct, for a 42% precision.

These results indicate that the system creates perfect clusters in about 40% of the case. Since the rating of cluster quality can take many levels between perfect and totally bad (e.g., a cluster may have four related terms and an unrelated one, and would still be judged bad in this evaluation), the true quality of the clustering component of ONTOSTRUCT in terms of precision is higher. The results of the clustering evaluation are also more impressive if we consider the noise present in the input data (via imperfections in term and relationship extraction, for example). This makes a strong point for the algorithm's ability to learn regularities in the data and to discriminate against artificially introduced examples.

4 Conclusions and Future Work

ONTOSTRUCT shows that full learning of ontological information from the basic terms to conceptual links is feasible, and introduces several new techniques for handling text data with partial structure, even when the available data is much smaller than the data sets typically used in data or text mining. We are currently designing a more extensive evaluation of ONTOSTRUCT's performance in this domain, bringing in domain experts from New Jersey state government and devising techniques for approximate measuring of the system's recall. We are also planning to apply ONTOSTRUCT on another domain where a gold standard is available, to study the generality of the approach and also compare directly with other semi-automated or automated approaches for some of the tasks that ONTOSTRUCT performs.

A text mining technique that would complement ONTOSTRUCT's extraction capabilities would be the automatic learning of relationship patterns; currently, ONTOSTRUCT relies on predefined syntactic patterns for this sort of data mining. We are currently implementing techniques for learning additional patterns and adding them into ONTOSTRUCT's operation. We are also looking into the issue of merging information

extracted from different documents or different sources which may be conflicting, and rating the confidence of ONTOSTRUCT's output on the basis of evidential measures for a piece of information across data sources. Finally, ONTOSTRUCT output is being combined with a workflow management system developed at Rutgers University for obtaining an online system that can be used for facilitating business registration in New Jersey.

Acknowledgments

We wish to thank Nabil Adams, Soon ae Chun, and the other members of the E-Government research team for the New Jersey business registration portal, for useful suggestions and comments during the development of this work. The work is supported in part by National Science Foundation grant EIA-9983468 under the Digital Government program. Any opinions, findings, or recommendations are those of the authors and do not necessarily reflect the views of the NSF.

References

N. Adams, F. Artigas, V. Atluri, S. A. Chun, S. Colbert, M. Degeratu, A. Ebeid, V. Hatzivassiloglou, R. Holowczak, O. Marcopolus, P. Mazzoleni, W. Rayner, and Y. Yesha. E-Government: Human-Centered Systems for Business Services. In *Proceedings of the First National Conference on Digital Government Research*, 2000.

E. Agichtein and L. Gravano. Snowball: Extracting Relations from Large Plain-Text Collections. In *Proceedings of the Fifth ACM International Conference on Digital Libraries*, 2000.

G. Bisson, C. Nédellec, and D. Cañamero. Designing Clustering Methods for Ontology Building: the Mo'K Workbench. In *Proceedings of the First Workshop on Ontology Learning (OL-2000) in conjunction with the Fourteenth European Conference on Artificial Intelligence (ECAI-2000)*, Berlin, 2000.

S. Finch and A. Mikheev. A Workbench for Finding Structure in Texts. In *Proceedings of the Fifth Conference on Applied Natural Language Processing*, April 1997.

J. S. Justeson and S. M. Katz. Technical terminology: Some Linguistic Properties and an Algorithm for Identification in Text. *Natural Language Engineering*, **1**(1):9–27, 1995.

K. Knight and S. Luk. Building a Large Knowledge Base for Machine Translation. In *Proceedings of the American Association of Artificial Intelligence Conference (AAAI-94)*, 1994.

G. N. Lance and W. T. Williams. A General Theory of Classification Sorting Strategies. *Computer Journal*, **9**:373–380, 1967.

D. B. Lenat. CYC: A Large-Scale Investment in Knowledge Infrastrature. *Communications of the ACM*, **38**(11):32–38, 1995.

G. A. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. J. Miller. Introduction to WordNet: An On-Line Lexical Database. *International Journal of Lexicography*, **3**(4):235–312, 1990.

M. F. Porter. An Algorithm for Suffix Stripping. *Program*, **14**(3):130–137, 1980.

J. C. Reynar and A. Ratnaparkhi. A Maximum Entropy Approach to Identifying Sentence Boundaries. *Proceedings of the Fifth Conference on Applied Natural Language Processing*, April 1997.

M. Sanderson and B. W. Croft. Deriving Concept Hierarchies from Text. In *Proceedings of the 22nd ACM SIGIR*, pages 206–213, 1999.

S. Siegel and N. J. Castellan. *Nonparametric Statistics for the Behavioural Sciences*. McGraw-Hill, 2nd edition, 1988.