OntoSoar: Using Language to Find Genealogy Facts

Proposal for a Thesis for the Degree of MA in Linguistics
Peter Lindes
Brigham Young University
Fall 2013

> *Thus, intelligence is the ability to bring to bear all the knowledge that one has in service of one's goals.*
>
> *Newell (1990), p. 90*

Abstract

There is a need to have an automated system that can read genealogy books or other historical texts and extract as many facts as possible from them. Embley and others have applied traditional information extraction techniques to this problem with a reasonable amount of success. In parallel much linguistic theory has been developed in the past decades, and Lonsdale and others have built computational embodiments of some of these theories. The goal of this thesis is to apply computational models of linguistic theory to the problem of extracting facts from genealogy books to find facts that traditional information extraction techniques could not find.

Introduction

An enormous amount of unstructured and semi-structured text is available in many domains, containing huge quantities of human readable information. A way of making this information machine readable so that it can be easily queried is a much sought after goal.

One approach to the problem is in a large literature on information extraction from text. Some samples of this literature include Buitelaar et al (2009), Carlson et al (2009), Cimiano (2006), Nguyen et al (2008), Pivk et al (2005), Tao and Embley (2007), Tijerino et al (2003), Volker et al (2007), Wong et al (2012), Yao and Hamilton (2007), and Yang et al (2008).

Embley and others have applied these techniques to build a system called OntoES (Embley et al (1999 and 2011)), which has been used to extract facts from a variety of text categories, including internet car ads, obituaries, and family history books. The system has been adapted to work in several languages, including English, French, and Korean, and to be able to query data across languages. However, the ability of OntoES (and similar systems) to extract information from unstructured and semi-structured natural language text is limited by its inability to understand the complex syntactic and semantic structures of natural language.

In parallel with this progress in information extraction there has been over the last several decades a tremendous growth in linguistic theory that can explain syntactic, semantic, and other linguistic phenomena over a wide range of the world's languages. In recent years this has culminated in Chomsky's Minimalist Program

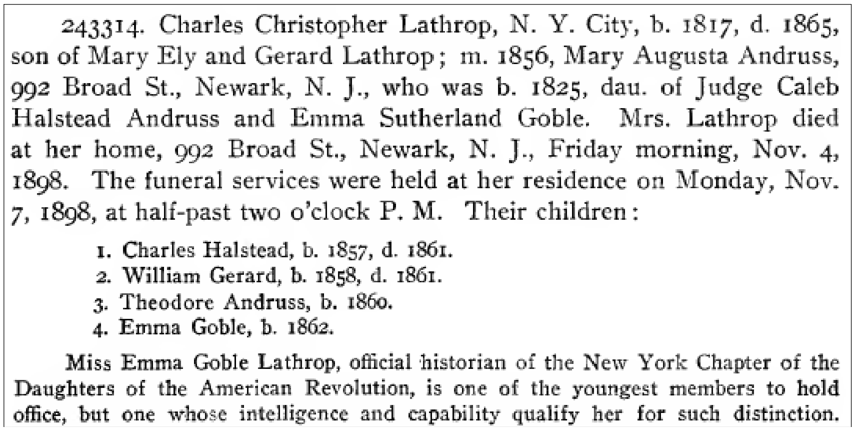
(Chomsky (1995)) for syntax and Jackendoff's theory of Conceptual Semantics (Jackendoff (1990, 1996, 2002, and 2003)). However, there has not been a great deal of practical application of these linguistic theories to building information extraction systems.

Lonsdale and others have pioneered in this area by applying the Soar cognitive architecture to build the LG-Soar and XNL-Soar systems (Lonsdale et al (2001), Lonsdale et al (2007), and Lonsdale et al (2012)). Melby (1995a, 1995b) has also shown the necessity of having a cognitive agent, such as Soar, to be able to achieve machine understanding of natural language.

The goal of this thesis is to combine these two threads of research into a single system we call OntoSoar, and to demonstrate its usefulness by applying it to extracting information from family history books. We hope to show that OntoSoar can find facts that OntoES alone could not.

Examples of text

Before getting into the details of the OntoSoar system, let's look at the problem in a little more detail. Figure 1 shows part of an image of page 419 of *The Ely Ancestry*, while Figure 2 is a section of page 84 of *A Genealogical History of the Harwood Families, Descended from Andrew Harwood*.

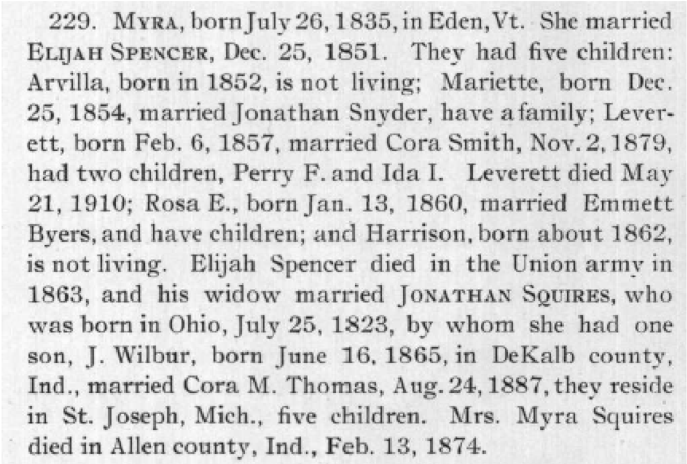


243314. Charles Christopher Lathrop, N. Y. City, b. 1817, d. 1865, son of Mary Ely and Gerard Lathrop; m. 1856, Mary Augusta Andruss, 992 Broad St., Newark, N. J., who was b. 1825, dau. of Judge Caleb Halstead Andruss and Emma Sutherland Goble. Mrs. Lathrop died at her home, 992 Broad St., Newark, N. J., Friday morning, Nov. 4, 1898. The funeral services were held at her residence on Monday, Nov. 7, 1898, at half-past two o'clock P. M. Their children:

1. Charles Halstead, b. 1857, d. 1861.
2. William Gerard, b. 1858, d. 1861.
3. Theodore Andruss, b. 1860.
4. Emma Goble, b. 1862.

Miss Emma Goble Lathrop, official historian of the New York Chapter of the Daughters of the American Revolution, is one of the youngest members to hold office, but one whose intelligence and capability qualify her for such distinction.

Figure 1: A Sample of Genealogy Text



229. Myra, born July 26, 1835, in Eden, Vt. She married Elijah Spencer, Dec. 25, 1851. They had five children: Arvilla, born in 1852, is not living; Mariette, born Dec. 25, 1854, married Jonathan Snyder, have a family; Leverett, born Feb. 6, 1857, married Cora Smith, Nov. 2, 1879, had two children, Perry F. and Ida I. Leverett died May 21, 1910; Rosa E., born Jan. 13, 1860, married Emmett Byers, and have children; and Harrison, born about 1862, is not living. Elijah Spencer died in the Union army in 1863, and his widow married Jonathan Squires, who was born in Ohio, July 25, 1823, by whom she had one son, J. Wilbur, born June 16, 1865, in DeKalb county, Ind., married Cora M. Thomas, Aug. 24, 1887, they reside in St. Joseph, Mich., five children. Mrs. Myra Squires died in Allen county, Ind., Feb. 13, 1874.

Figure 2: Sample 2 from Harwood 84

Here we see a portion of text rich in facts to be extracted. We also see a lot of linguistic complexity at the lexical, syntactic, and semantic levels. Lexically we have repeated abbreviations of *born* as *b.*, *died* as *d.* and *married* as *m.*. Syntactically we see a sentence like (1a) that to be a minimally grammatical sentence in normal English would have to be written something like (1b). The sentence would be even more natural if expressed as in (1c). The system will have to adapt any algorithms based on standard English to deal with this abbreviated form, which can be considered a domain-specific dialect of English.

(1) a. `Charles Halstead, b. 1857, d. 1861.`

b. `Charles Halstead, born in 1857, died in 1861.`

c. `Charles Halstead was born in 1857 and died in 1861.`

At the syntactic and semantic levels there are other issues to deal with. On the second line after the semicolon we see (2a), which would be more natural English if in the form of (2b) or (2c). Notice the need to find the antecedents for *he, who*, and *she.*

(2) a. `m. 1856, Mary Augusta Andruss, ... who was b. 1825, ...`

b. `In 1856 he married Mary Augusta Andruss, who was born in 1856 ...`

c. `He married Mary Augusta Andruss in 1856.  She was born in 1856 ...`

Then in (3) we see another kind of challenge. To figure out who was the person that died in this sentence we must be able to infer from the context that since *Mary Augusta Andruss* married *Charles Christopher Lathrop* she could also be known as *Mrs. Lathrop* and therefore she is probably the person who died.

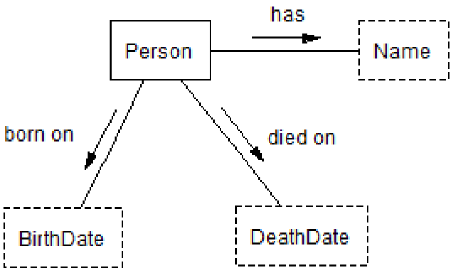
(3) `Mrs. Lathrop died at her home ...`

The small portions of text shown in Figures 1 and 2 have a number of riddles of this sort that need to be solved in order to achieve something close to human performance in discovering genealogical facts, even if we confine ourselves to identifying individuals, their birth and death dates, who they are married to, and who are their children.

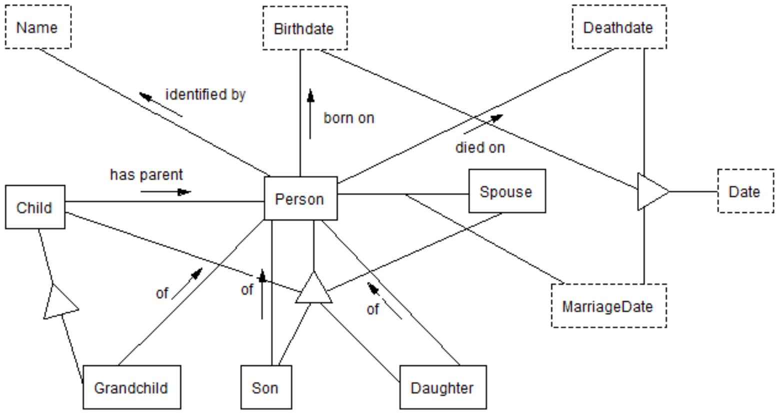
Example ontologies

In (4) we see two examples of conceptual models that might be used:

(4) a. Simple person/birth/death model.



b. Complete family model



Clearly a fairly deep understanding of the domain-specific language involved, along with the power to do sophisticated inferencing, will be required to fully solve this problem. OntoSoar will certainly fall short of human performance, but we expect it to be able to do a good deal better than the previous OntoES system alone. And it should be a good basis for further development in the future.

Thesis Statement

The primary hypothesis we hope to prove with this thesis is the following:

(5) *We can develop an algorithm to match data extracted from text using modern lexical, syntactic, and semantic analysis tools to a conceptual model of a domain provided by a user so as to populate the model with facts found in the text, and that such a system can find facts that traditional information extraction systems could not.*

Method

Figure 3 shows a block diagram of the OntoSoar system. The core of the system is the row of blocks in the center that start with the text extracted from a PDF with OCR and process it through several stages to produce a populated ontology in the form of an OSMX file.

In Figure 3 the OntoES system servers three important purposes: it manages PDF files and the OCR extraction of text from them, it provides tools for a user to build a conceptual model of the domain in the form of an OSMX file called an *ontology*, and it provides tools to visualize the resulting populated ontology and compare the facts found by OntoSoar with ones found by either a human annotator or other OntoES automated processes.

The main OntoSoar process is managed by a Java program not explicitly shown in Figure 3. This program inputs a page or paragraph of text at a time, runs it through a segmenter to divide the raw text into sentences of sentence fragments, runs each segment into the LG Parser to perform syntactic analysis, and then submits the syntax linkage to Soar, where semantic processing takes place in several stages.

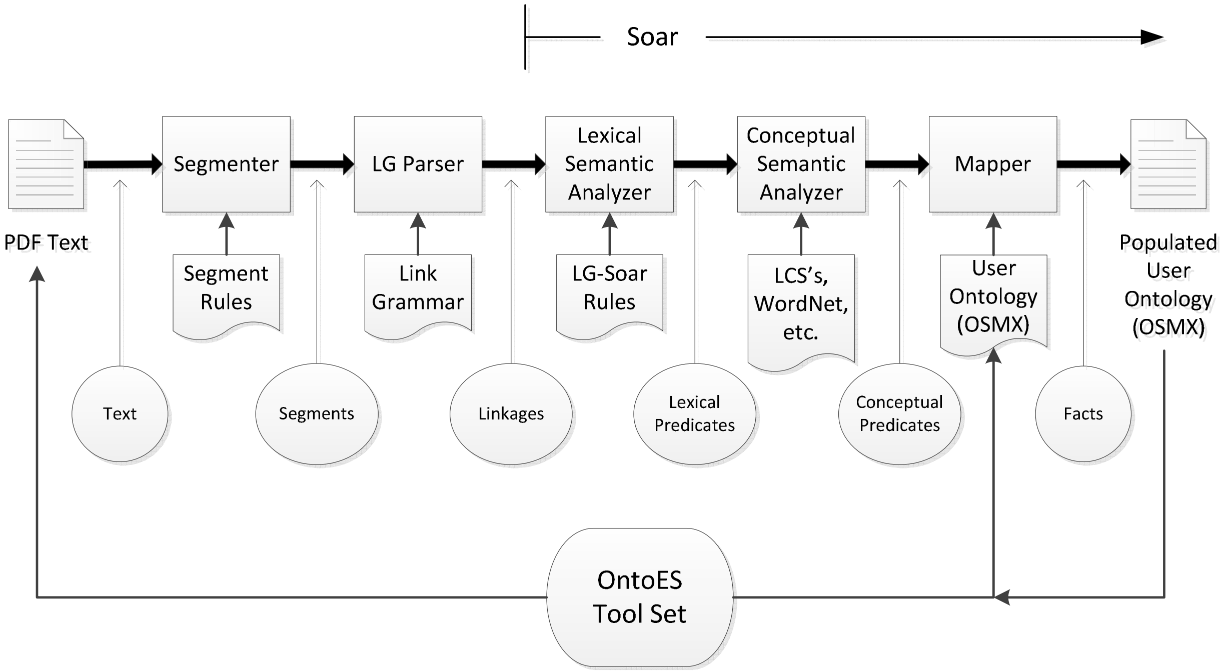


Figure 3: OntoSoar Block Diagram

LG Parser

The syntactic analysis component of OntoSoar is built on the Link Grammar Parser (Sleater and Temperly (1991 and 1993), Lafferty, Sleator, and Temperly (1992), and Grinberg, Lafferty, and Sleator (1995)). Code is available at http://www.abisource.com/projects/link-grammar/.

This parser was chosen for several reasons: it is quite robust in dealing with many English grammatical structures, its grammar is readily adapted by modifying text files, and it has already been integrated into LG-Soar and used for a number of other projects (eg. Lonsdale et al. (2007), Parker (2005), Tustison (2004), and Wintermute (2012)).

Semantic analysis

The semantic analysis in Soar begins by building a simple semantic representation of the elements in the syntactic linkage, as has been previously done with LG-Soar (see Lonsdale et al. 2001, 2007, and 2012). Next a conceptual analyzer identifies verbs and their arguments, names, dates, etc.

Discourse analysis

Although it is not shown explicitly as a block in Figure 3, an important component of the semantic analysis is *discourse analysis*. This consists of keeping trap over multiple segments, at least on the scale of a paragraph, what referents are salient and using this knowledge to resolve anaphoric expressions.

To illustrate the issues involved, consider the text in Figures 2. The pronouns *she* and *they* appear several times, as well as anaphoric referring expressions such as *his widow* and *Mrs. Myra Squires*. In Figure 1 we have anaphora such as *Mrs. Lathrop* and *her*. Both examples have headings for lists of children, but the parent-child relationships are not made specific. As part of this thesis we expect to include a discourse analysis component which should be able to resolve at least some of these anaphora.

Mapper

The core of this thesis is the module, called the Mapper in Figure 2, which finds matches between linguistic information in the conceptual predicates and the elements of the user ontology and uses these matches to populate the ontology with facts. The following paragraphs will describe this matching algorithm in detail, along with showing the details of its operation on a simple example.

Goal

(6) *Match semantic data derived from text using a linguistic model to a conceptual model (or ontology) for a specific domain prepared by a user so as to populate the model with facts.*

Desired facts

(7) a. Individual persons identified by their names.

b. Birth and death dates for these individuals.

c. Marriages between individuals.

d. Parent/child relationships between individuals.

e. Other derived relationships such as *grandchild.*

Data from linguistics

(8) a. Names, dates, and other *referring expressions*, including pronouns and descriptions such as *John's mother*.

b. Entities derived from the referring expressions, mainly people but possibly places, institutions, etc.

c. Verbs and their arguments.

d. Relationships between entities derived from the verbs.

e. Other relationships derived from phrases such as *son of* or *lived in.*

Data from the conceptual model

We can define four possible levels of information to be provided in the user ontology, as shown in (9). We wish to determine how the choice of ontology level affects the number of facts we can derive.

(9) a. Level 1 - Object sets and their names only, along with object existence rules.

b. Level 2 - Add relationships sets with linguistically meaningful names.

c. Level 3 - Add individual entities derived from the text by OntoES that are members of the given object sets.

d. Level 4 - Add instances to the relationship sets derived from the text by OntoES.

Matching algorithm

We want an algorithm that will take data as described in (8), map it to data like that in (9), and produce facts as described in (7). There are probably several ways to approach this problem, but the one described here will be centered on first matching verbs to relationship sets and then using the arguments of both to derive the facts needed.

For the moment let's consider a fairly simple case where the ontology looks like the one in (4a) and the input sentence is of the form shown in (10).

(10) <subject> <verb> <date>

This is a very simple sentence structure that describes an event that happened in the life of some person. Such sentences are very common in or domain. The algorithm will proceed as shown in (11).

(11) a. Step 1 – Take the root verb of the sentence and match to a relationship set in the ontology by looking for a match between the verb and the name of the relationship set.

b. Step 2 – Identify the entity that is the subject of the verb as a member of the object set which is the subject of the verb's relationship set.

c. Step 3 – Enter the name of the subject entity as an instance of the lexical object set which names the entity's object set.

d. Step 4 – Enter the date attached to the verb as a member of the lexical object set which is the object of the verb's relationship set.

Although this algorithm is quite limited, it serves to show the general approach we can use. In the next section we will look at how it applies in detail to a specific example, and then we will consider various alternatives and ways the algorithm could be extended to handle more general cases.

A simple example

The following shows the processing performed by the existing OntoSoar code base on a sample sentence.

(12) a. `Mary died in 1853.`

```
b.                                  c.                   d.
        +------------Xp-----------+      in(died,N4)          VERB(N3,"died")
        +---Wd--+--Ss-+-MVp+-IN+  |      1853(N4)             Subject(N3,N2)
        |       |     |    |   |  |      Mary(N2)             NAME(N6,"Mary")
                                         died(N2)             DATE(N4,"1853")
    LEFT-WALL Mary died.v in 1853 .                           happened(E6,N4)
                                                              named(N3,N4)
```

Here (12a) shows the input sentence, (12b) the output of the LG parser, and (12c) the predicates produced by the traditional LG-Soar semantics logic. The predicates in column (12d) were generated by the new conceptual semantics processing in OntoSoar.

Now we can look at what the Mapper does. It has two inputs, the semantic predicates shown in (12d) and the user ontology summarized in (13a). Through a series of steps it produces the matches and facts shown in (13b).

The *MATCH* predicates in (13b) are the key here. The first one says it has found a match between *N3*, which is the verb *died* in this sentence, and *R20* which is the id assigned in Soar for the *died on* relation from the ontology. (The user ontology being used here is essentially the simple one shown in (4a).) Similarly, the *NAME* predicate is matched to the *Name* lexical object set called *O23, named* matches *R18 has*, *DATE* is matched to *O25 DeathDate*, and *happened* to the object side of *R20 died on*.

(13) a. Object Sets
O22 osmx3 *Person* oe-rule O26
O23 osmx7 *Name**
O24 osmx20 *BirthDate**
O25 osmx5 *DeathDate**
Object Existence Rules
O26: O22 -> {*Name*}
Relationship Sets
R18 osmx16 O22 *has* O23
R19 osmx23 O22 *born on* O24
R20 osmx10 O22 *died on* O25

b.

```
    MATCH(N3,R20)
[X1:O22] Person(X1), MATCH(N2,X1)
    MATCH(NAME,O23)
    MATCH(named,R18)
[X2:O23] Name(X2,"Mary"), MATCH(N6,X2)
[Y1:R18] Person(X1) has Name(X2)
    MATCH(DATE,O25)
    MATCH(happened,R20*)
[X3:O25] DeathDate(X3,"1853"),
              MATCH(N4,X3)
[Y2:R20] Person(X1) died on DeathDate(X3)
```

Based on these matches, we create new individuals *X1*, *X2*, and *X3* in the ontology as members of *Person*, *Name*, and *DeathDate* respectively, along with instances *Y1* and *Y2* of the *has* and *died on* relationship sets. These id's will be converted to osmx numbers when the populated ontology is output from the system.

Alternative algorithms

The verb-centered algorithm presented above is basically a top-down algorithm. It depends completely on having meaningful names for some relationship sets and lexical object sets. Another sort of algorithm, working bottom up from the lexical items, would be another possibility. However, it presents some challenges.

Consider that we have used linguistic knowledge to determine that a phrase is a proper name. How do we know if it is the name of a person, a place, an organization, or something else? A bottom-up approach depends on knowing the type of the entity. Determining this would require using dictionaries or external NER tools. In our example ontologies knowing that something is a date still leaves ambiguous what kind of date it is.

Another issue is how to correlate non-lexical object sets with entities. The names of these object sets do not correlate to any linguistic cues, so we would need some additional correlation table to be provided externally.

In this thesis we will concentrate primarily on the top-down algorithm since this is the one which takes the most advantage of the unique linguistic knowledge of OntoSoar. Bottom-up algorithms as described here will need to be left for future work.

Evaluation

Since OntoSoar is a novel approach to extracting genealogical information from family history texts, there is no similar system to directly compare it with. Nevertheless, we can measure its performance in a number of alternative ways, as outlined here.

Basic evaluation

The basic evaluation will be to build the system, optimize it as much as possible, and compare its results on various test paragraphs to human annotation, measuring precision and recall. For this basic evaluation we will use paragraphs such as those shown in Figures 1 and 2, and ontologies like those given in (4a) and (4b). During development we will continuously work to optimize the results for these test cases.

Component evaluation

The core of this thesis is the Mapper component which matches semantic structures to the given ontology and populates the ontology with facts. The success of this mapping will depend on the quality of the semantic data provided to it. Several other components of the system can cause problems in this semantic input: OCR, segmentation, parsing, and semantic analysis.

In order to judge the accuracy of the Mapper itself, we can modify the intermediate results of these other stages of processing to see what happens when we manually correct errors in OCR, parsing, etc. In this way we can get a better idea of the performance of each component by itself.

Ontology exploration

Another important type of evaluation will be to try different variations of the input ontologies. Variations both in structure and linguistic content will be important to understand what is needed from the ontology in order for OntoSoar to perform well.

Quantity evaluation

The basic evaluation will be done on the texts shown above, and this will drive development. Once the system is working well, we can run it on a much larger data set and pick random parts of that data set to evaluate compared to human annotation.

Comparisons

It would be interesting to compare OntoSoar with other OntoES tools that extract the same kind of facts, perhaps Frontier for example, in order to see how they compare and discover in what kinds of situations OntoSoar is better than or worse than other techniques.

Project Plan

The project of completing this thesis is planned to proceed according to the following estimates:

Phase	Description	Plan A	Plan B
Committee Review	Committee meeting to review proposal document	9 May 2013	9 May 2013
Approved Proposal	A final proposal document approved	4 Oct 2013	4 Oct 2013
Working System	OntoSoar works as a whole to produce facts from example texts and ontologies	15 Oct 2013	8 Nov 2013
Evaluation	Studies completed to evaluate the system in the several ways described above	30 Oct 2013	29 Nov 2013
Draft Thesis	A complete draft of the written thesis delivered to committee members	1 Nov 2013	17 Dec 2013
Schedule Defense	Final oral examination date set with approval of committee members	15 Nov 2013	15 Jan 2014
Defense	Final oral examination	29 Nov 2013	30 Jan 2014
Submission	Approved thesis submitted to college dean	10 Dec 2013	14 Feb 2014
Grad Studies	ETD submitted and ADV Form 8d taken to Graduate Studies	17 Dec 2013	28 Feb 2014

Conclusions

OntoSoar is a step into the realm of linguistically sophisticated systems for extracting information from historical documents. It will certainly not solve all the problems or answer all the interesting questions in this field. It should, however, be able to show that such an approach can produce substantial benefits, and point the way for more advanced approaches.

Bibliography

Carlson, Andrew, Justin Betteridge, Estevam R. Hruschka Jr., and Tom Mitchell (2009). "Coupling Semi-Supervised Learning of Categories and Relations" in *Proceedings of the NAACL HLT 2009 Workshop on Semi-supervised Learning for Natural Language Processing*, Boulder, Colorado, June 2009, pp. 1-9.

Buitelaar, Paul, Philipp Cimiano, Peter Haase, and Michael Sintek (2009). "Towards Linguistically Grounded Ontologies," *Proceedings of the 6th European Semantic Web Conference (ESWC'09)*, Heraklion, Greece, May/June 2009.

Chomsky, Noam (1995). A Minimalist Program for Linguistic Theory. In N. Chomsky *The Minimalist Program* (pp. 167-217). Cambridge MA, MIT Press.

Cimiano, Phlipp (2006). *Ontology Learning and Population from Text: Algorithms, Evaluation and Applications*. Springer, New York.

Embley, D. W., D. M. Campbell, Y. S. Yiang, S. W. Liddle, D. W. Lonsdale, Y.-K. Ng, and R. D. Smith, (1999). *Conceptual-Model-Based Data Extraction from Multiple-Record Web Pages*. Available on the CS 652 Fall 2012 web site.

Embley, David W., Steven W. Liddle, and Deryle W. Lonsdale, (2011). “Conceptual Modeling Foundations for a Web of Knowledge”, in *Handbook of Conceptual Modeling*, Chapter 15. Available on the CS 652 Fall 2012 web site.

Grinberg, Dennis, John Lafferty, and Daniel Sleator (1995). A robust parsing algorithm for link grammars. Carnegie Mellon University Computer Science technical report CMU-CS-95-125, and *Proceedings of the Fourth International Workshop on Parsing Technologies*, Prague, September, 1995.

Jackendoff, Ray (1990). *Semantic Structures*. The MIT Press.

Jackendoff, Ray (1996). "Semantics and Cognition," in Shalom Lappin ed. *The Handbook of Contemporary Semantic Theory*. Blackwell.

Jackendoff, Ray (2002). *Foundations of Language: Brain, Meaning, Grammar, Evolution*. Oxford University Press.

Jackendoff, Ray (2003). Précis of *Foundations of Language: Brain, Meaning, Grammar, Evolution*. *Behavioral and Brain Science*s, 26, 651-707.

Lafferty, John, Daniel Sleator, and Davy Temperley (1992). Grammatical Trigrams: A Probabilistic Model of Link Grammar. *Proceedings of the AAAI Conference on Probabilistic Approaches to Natural Language*, October, 1992.

Laird, John E. (2012). *The Soar Cognitive Architecture*. The MIT Press, Cambridge, MA.

Lonsdale, Deryle, Merrill Hutchison, Tim Richards, and William Taysom (2001). *An NLP system for extracting and representing knowledge from abbreviated text*. Deseret Language and Linguistics Symposium, 2001.

Lonsdale, D. W., C. Tustison, C. G. Parker, and D. W. Embley (2007). "Assessing clinical trial eligibility with logic expression queries," *Data & Knowledge Engineering,* Volume 66 Issue 1, July, 2008, Pages 3-17.

Lonsdale, Deryle, David W. Embley, and Steven W. Liddle, (2012). *An ontology-driven reading agent*. Available 9/30/2013 at http://www.deg.byu.edu/proposals/readingagent.pdf.

Melby, Alan K. (1995a). *Why Can't a Computer Translate More Like a Human*? 1995 Barker Lecture, available at http://www.ttt.org/theory/barker.html.

Melby, Alan K. and C. Terry Warner (1995b). *The Possibility of Language: A Discussion of the Nature of Language, with Implications for Human and Machine Translation.* John Benjamins Publishing Company, Amsterdam/Philadelphia.

Newell, Alan (1990). *Unified Theories of Cognition.* Cambridge, MA: Harvard University Press.

Nguyen, Hoa, Thanh Nguyen, and Juliana Freire (2008). "Learning to Extract Form Labels," *Proceedings of the 34th International Conference on Very Large Data Bases (VLDB'08)*, Auckland, New Zealand, August 2008, 684-694.

Parker, Craig G. (2005). *Generating Medical Logic Modules for Clinical Trial Eligibility.* BYU CS Master's Thesis, November, 2005.

Pivk, Aleksander, York Sure, Philipp Cimiano, Matjaz Gams, Vladislav Rajkovic, and Rudi Studer (2005). "Transforming Arbitrary Tables into F-Logic Frames with TARTAR," *Data & Knowledge Engineering*, 60, 2007, 567-595.

Sleator, Daniel D. K. and Davy Temperley (1991), *Parsing English with a Link Grammar*, Carnegie Mellon University Computer Science technical report CMU-CS-91-196, October 1991.

Sleator, Daniel D. K. and Davy Temperley (1993), Parsing English with a Link Grammar, *Third International Workshop on Parsing Technologies.*

Tao, Cui and David W. Embley, (2007). Automatic Hidden-Web Table Interpretation by Sibling Page Comparison. *Lecture Notes in Computer Science*, Volume 4801, 2007, pp. 556-581.

Tijerino, Yuri, David W. Embley, Deryle W. Lonsdale, and George Nagy, (2004). *Towards Ontology Generation from Tables.* Available 9/30/2013 at http://http://tango.byu.edu/papers/Tango20040430x.pdf.

Tustison, Clint A. (2003). "LG-Soar" in*The 23rd Soar Workshop*, June 2003, University of Michigan.

Tustison, Clint A. (2004). *Logical Form Identification for Medical Clinical Trials.* BYU Linguistics MA Thesis, December, 2004.

Völker, Johanna, Pascal Hitzler, and Philipp Cimiano (2007). "Acquisition of OWL DL Axioms from Lexical Resources," *Proceedings of the 4th European SemanticWeb Conference (ESWC 2007)*, Innsbruck, Austria, June 2007 670-685.

Wintermute, Sam (2012). “Leveraging Cognitive Context for Language Processing in Soar” in *The 32nd Soar Workshop*, June 2012, University of Michigan. Available at: http://ai.eecs.umich.edu/soar/sitemaker/workshop/32/.

Wong, Wilson, Wei Liu, and Mohammed Bennamoun (2012). "Ontology Learning from Text" in *ACM Computing Surveys*, Vol. 44, No. 4, Article 20.

Yang, Shao-Hua, Hai-Lue Lin, and Yan-Bo Han (2008). "Automatic Data Extraction from Template-generated Web Pages," *Journal of Software*, 19(2), 209-223, 2008.

Yao, Hong, and Howard J. Hamilton (2008). "Mining Functional Dependencies from Data," *Data Mining and Knowledge Discovery*, 16, 2008, 197-219.