DEKADE: An Environment Supporting Development of NLP Systems

Jesse English and Sergei Nirenburg
Institute for Language and Information Technologies
University of Maryland, Baltimore County
jesse.english@umbc.edu, sergei@umbc.edu

Abstract

This paper describes ongoing work on the DEKADE (Development, Evaluation, Knowledge Acquisition, and Demonstration Environment) system and its components, the DekadeAPI, the DekadeServer, and the DekadeClient. DEKADE supports the development and operation of the natural language processing (NLP) system OntoSem, including its processors and static knowledge resources as well as applications that rely on OntoSem for their natural language processing needs

1. Introduction

Automatic extraction of meaning from unstructured natural language text is, in some sense, the core capability underlying semantic computing. This paper describes some aspects of our ongoing work on a set of tools facilitating the development of a battery of processing modules and knowledge resources that together comprise the semantic analyzer called OntoSem [15]. The complexity of the knowledge involved in OntoSem processing, as well as the manifold interaction of its various modules makes the development and testing of the system impossible without sophisticated efficiency-enhancing tools.

Such tools must facilitate comprehensive testing of any modifications to the system's code by examining the results of several analyzer modules. In particular, allowing the developers to adjust the parameters of the execution at intermediate steps of text analysis (a capability similar to a typical code debugging interface) facilitates development of modules in arbitrary order, which is a desirable feature. Similarly, knowledge acquirers must be able to test the quality of newly added knowledge (e.g., ontological concepts or lexicon entries) by running the analyzer with the augmented static knowledge resources.

The adequate set of tools for supporting knowledge-based natural language processing must, of course, include a variety of knowledge editors. Availability of interactive editors for both the static knowledge resources and the results of the various processing modules (including the final output of OntoSem, text meaning representations, or TMRs) is essential. A good example of the utility of editing system results is the production of "gold standard" TMRs by having human users correct and augments the results produced automatically by the system. Gold standard TMRs have a number of uses in evaluating development progress and quality of the results as well as in creating a corpus of rich semantic representations of text meaning that can be used to train a variety of statistical models for semantic text analysis.

To be truly efficiency-enhancing, the interactive knowledge acquisition facilities in the tool set must facilitate automatic validation of the newly acquired knowledge elements (verifying that they are both syntactically and semantically sound), as well as allow the user to see how the various static knowledge resources interact.

Finally, the tool set must support the use of the OntoSem analysis environment by users who are not developers and those who want to incorporate OntoSem in their application.

To address the above issues, we have developed DEKADE, a Development, Evaluation, Knowledge Acquisition, and Demonstration Environment of OntoSem. DEKADE targets the developer, knowledge acquirer, and researcher requirements in a user-friendly, cross-platform, client-server solution.

2. OntoSem

OntoSem (the implementation of the theory of Ontological Semantics) is a text-processing

environment that takes as input unrestricted raw text and carries out preprocessing, morphological analysis, syntactic analysis, and semantic analysis, with the results of semantic analysis represented as formal text-meaning representations that can then be used as the basis for many applications. TMRs have been used as the substrate for question-answering (e.g., [4]), machine translation (e.g., [3]) and knowledge extraction, and were also used as the basis for reasoning in the question-answering system AQUA, where they supplied knowledge to showcase temporal reasoning capabilities of the reasoning system JTP [8]. Text analysis relies on the following static knowledge resources:

- The OntoSem language-independent **ontology**, which currently contains around 8,500 concepts, each of which is described by an average of 16 properties. The ontology is populated by concepts that we expect to be relevant cross-linguistically. The current experiment was run on a subset of the ontology containing about 6,000 concepts.
- An OntoSem **lexicon** whose entries contain syntactic and semantic information (linked through variables) as well as calls for procedural semantic routines when necessary. The current English lexicon contains approximately 30,000 senses, including most closed-class items and many of the most frequent and polysemous verbs, as selected through corpus analysis. The base lexicon is expanded at runtime using an inventory of lexical (e.g., derivational-morphological) rules.
- An **onomasticon**, or lexicon of proper names, which contains approximately 350,000 entries.
- A **fact repository**, which contains "remembered instances" of ontological concepts. The fact repository is not used in the current experiment but will provide valuable semantically-annotated context information for future experiments.
- The OntoSem syntactic-semantic **analyzer**, which performs preprocessing (tokenization, named-entity and acronym recognition, etc.), morphological, syntactic and semantic analysis, and the creation of TMRs.
- The TMR language, which is the **metalanguage** for representing text meaning (a converter was developed between this custom language and OWL, see [10]).

OntoSem knowledge resources have been acquired by trained acquirers using a broad variety of efficiency-enhancing tools – graphical editors, enhanced search facilities, capabilities of automatically acquiring knowledge for classes of entities on the basis of manually acquired knowledge for a single representative of the class, etc.

3. Related Work

A large number of tools have been developed in the field of NLP over the years, many of them devoted to raising the efficiency of knowledge acquisition [e.g. 1, 12, 9, 13, 6, to name a few systems]. In this paper, we will briefly review a small subset of such tools selected from among those whose goals, coverage or architecture has similarities with DEKADE.

The core purpose of FrameNet [1] is to facilitate semantic annotation of text and lexicon. The tool used to support this was originally a dynamic web environment. Using Perl/CGI, the interface was created by gluing together off-the-shelf software used to communicate with the data structure of FrameNet's frames. Later, an API was developed [2], along with a series of desktop tools, which were combined to make for a more intuitive user platform. In contrast with DEKADE, the FrameNet tools do not need to support the development of an automatic semantic analyzer and therefore does not need to conform to the latter's specifications.

ConceptNet [12] is a toolkit supporting the applications of topic gisting, text summarization, affect-sensing and some others and associated with a "commonsense knowledge base" that uses an ontological metalanguage that can be characterized as constrained English. The data set ConceptNet was created on the Open Mind Common Sense Project [17], a project where non-expert volunteers from across the web were asked to provide common-sense data. ConceptNet is backed by an NLP system, which is used to provide greater flexibility to the researcher using the available knowledge base by allowing access to a variety of commonsense NL functions. Unlike DEKADE, ConceptNet does not aim to provide tools for the knowledge acquirer (the knowledge comes from the Open Mind Common Sense Project); however, similar to DEKADE, ConceptNet makes the use of its NL tools easy for the researcher.

The Annotation Graph Toolkit (AGT) [13] provides a similar service as FrameNet, but with a different clientele in mind. AGT specifically targets annotation of time-series data, and provides an API for constructing tools that facilitate the construction of interfaces (using IDL [20]). AGT's aim is primarily to facilitate the knowledge acquisition process, and is flexible enough to allow the end-user to customize the interface. Similarly, DEKADE's knowledge modeling software, backed by a strong API, allows for flexible interface construction, if the standard interface does not fit the application.

Protégé [9] is an ontology development toolkit whose general methodology and design are closest to that of DEKADE. The Protégé system has been developed on the Java platform, with an open API to allow for easily created custom plug-ins to their tabbed environment. Protégé facilitates the knowledge acquisition process by supporting a method of validation within the framework of the interface. The API allows for the data acquired to be easily accessed in a platform-independent manner. Unlike DEKADE, Protégé does not directly support any NL system; instead Protégé's primary goal is as an ontology development platform, which NL researchers can then plug into.

The GATE environment [5, 6] focuses on streamlining the entire process of creating a NL system; it provides extensible tools and interfaces that facilitate the developer's task of crafting a language-based system from a variety of available resources. Linking in with resources such as WordNet [14], and Protégé, GATE allows the knowledge acquirer to define and specify various language resources such as lexicons and ontologies. An open API allows the researcher to access these tools and integrate them into an existing project, or construct a new one. GATE also integrates various machine-learning algorithms (via WEKA [19]), as well various evaluation-oriented algorithms.

Similar in scope to the GATE project, DEKADE aims to facilitate the tasks of the developer, knowledge acquirer, and researcher by providing an open framework API; however, DEKADE's focus is explicitly for the OntoSem environment, allowing the modification of even the most fundamental data-access methods. The tools suite for the developer and knowledge acquirer have been constructed with OntoSem in mind, and are then made available in a platform-independent way for the researcher.

4. Prior Work: Existing Tools for OntoSem

OntoSem has been under development for over 20 years. The last version of the core semantic analysis algorithms was developed in 1996. Since then a variety of tools have been constructed to assist in its development and in the acquisition of its static knowledge resources. Prior to the DEKADE system, using OntoSem for outside research required an in-depth, developer's view of the various modules and data repositories; thus, to obtain a full semantic analysis of a sentence one had to run scripts and programs written in Perl, C, C++, CLISP, and Java. Understanding the output meant having an intimate understanding of the TMRs produced, as well as the inner workings of the ontology and lexicon.

An early toolset designed specifically for OntoSem was KBAE (Knowledge Base Acquisition Editor). KBAE was designed as a web-based interface for ontology acquisition only (it did not support any other knowledge acquisition, nor did it support the development of OntoSem's processors, or semi-automatic production of TMRs). KBAE did offer a variety of useful features for ontology acquisition, most notably a very robust validation processor (a system that verified that the knowledge entered was not only syntactically correct, but also semantically correct given the current state of the knowledge). KBAE did have drawbacks: the data structure supported by KBAE was not the same as the one supported by OntoSem, leading to an inconvenient need to run a script after ontology acquisition in order to convert data formats to a standard representation. Further, KBAE limited certain aspects of acquisition (for example, reification of properties was not supported). Finally, the validator used by KBAE to guarantee the quality of knowledge acquisition was legacy software, making it considerably difficult to maintain or update.

In an attempt to address these drawbacks, as well as to introduce lexicon editing and support for human augmentation of automatic analyzer functions, the first implementation of DEKADE was constructed. This version, also a web-based application, was constructed using JSP/Java technologies as well as SQL for data storage solutions. At that time DEKADE supported an ontology browser (an editor on par with KBAE was never achieved), a lexicon editor (with fundamental UI aspects, as well as validation), and most notably an environment for processing a text through OntoSem.

This environment allowed the user to enter a text, and run each of the four major stages of textual analysis (preprocessing, syntactic, semantic, pragmatic/discourse), with an option of halting after each one for the user to inspect the intermediary output and adjust it if necessary. This feature was a significant asset, allowing the developer access to a convenient test-bench for the analyzer, and allowing the knowledge acquirer to quickly see how tweaking the static knowledge affected the final analysis. Each stage of the output was supported by a Java applet, allowing for a rich editing interface.

However, this first version of DEKADE also had its drawbacks: network lag, inconsistent HTML

rendering, and security were of major concerns to users. In addition, the ontology editor was still not up to par with existing resources, and there was a disconnect between the data stored by the DEKADE system, and the data used by OntoSem (at the time OntoSem did not access the database for its static knowledge resources, so updates to the flat files from the database had to be periodically run to keep the system in sync).

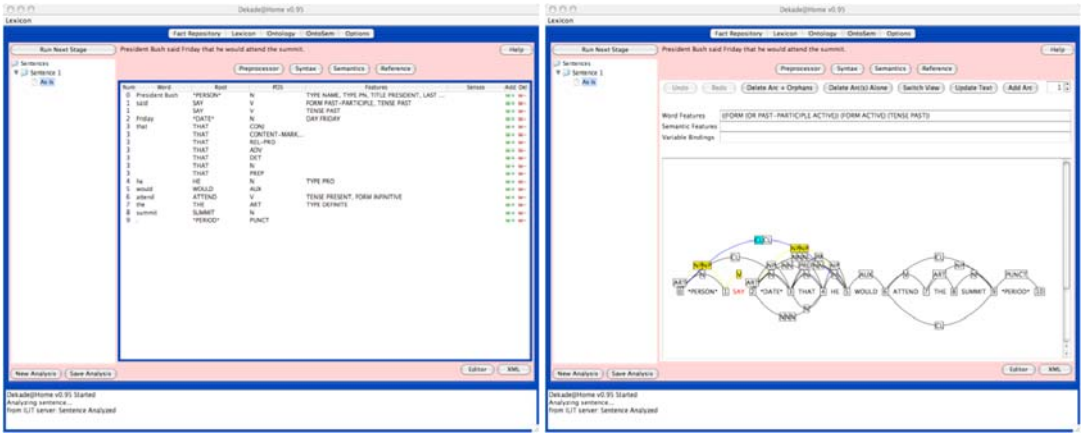
5. DEKADE Today

These problems led to the decision to create a new, robust, fully functional, integrated toolset. The current version of DEKADE abandoned the web interface for a custom, client-server architecture, built around an API designed to have uniform access to all of OntoSem's modules and static knowledge.

The initial task, and indeed the primary motivation to rebuild DEKADE from scratch, was to develop an open, and powerful, API. DekadeAPI, written in Java, is an extensible library that supports single function calls to access any of OntoSem's modules, as well as simple, yet robust, queries to the static knowledge resources. To improve the efficiency and coverage of these access methods, high-level Java objects have been created as wrapper classes to parse the results and present them to the user in an intuitive, easily accessible manner.

To further enhance the functionality of these methods, each was extended so that it is now possible to call it across an open-socket network connection, allowing the DekadeAPI to be usable by any user with an Internet connection. With this functionality available, it was time to construct an interface layer between the existing toolset and the user. The interface had to support the demands of three types of user: developer, knowledge acquirer, and researcher (who uses OntoSem as a tool), just as the DekadeAPI does.

Built on Java/Swing technology, the interface's parent UI frame handles securing the connection between itself (the DekadeClient), and the server application (the DekadeServer), and populates itself with a series of tabbed panes found in the application's root drag-and-drop folder, registering the browsing capabilities of each panel with the others. The interface was developed to support custom user panes that simply append to the interface and integrate with the existing tools. Using standard Java/Swing libraries, and custom DekadeAPI GUI extensions, researchers can easily



(a) (b)

(c)

Figure 1: OntoSem Stepped Analysis Interface

populate a panel with custom-built or existing DEKADE widgets, and use them for two-way communication with OntoSem.

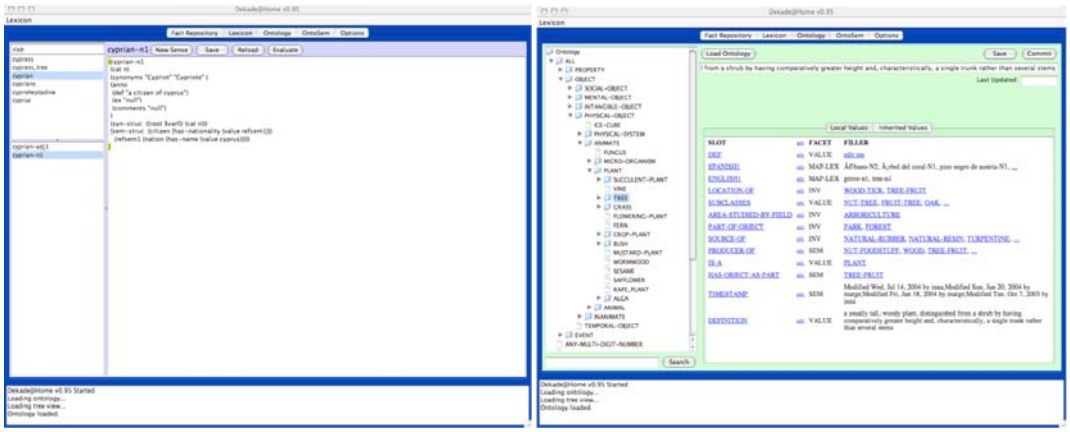
A key component, and a true improvement over many existing interfaces, is the interconnection between the various editors and browsers. In the new DEKADE environment, a knowledge acquirer can begin work on a lexicon entry, and in a single click inspect the corresponding ontological entry, and then swiftly return to the lexicon entry. The developer can also easily inspect the details of the various mappings to static knowledge made by any of the OntoSem processing modules to assist in the testing and debugging process of OntoSem.

The current standard version of the DekadeClient environment is supplied with the interfaces: to support OntoSem Stepped Analysis, Lexicon Browsing/Editing, Ontology Browsing/Editing, and Fact Repository Browsing/Editing.

5.1. OntoSem Stepped Analysis Interface

The OntoSem Stepped Analysis interface supports much the same functionality as the web-based DEKADE did, but with a higher level of interconnectivity and stability. The developer can enter text and then halt the analysis process at each of the four levels of analysis as required to inspect and, if necessary, modify the results. The interface supports three distinct stage editors, one for the preprocessor, (Fig. 1 (a)), one for the syntactic analyzer (Fig. 1 (b)) and one for the semantic and pragmatic/discourse analyzers (Fig. 1 (c)).

The preprocessor and syntax editor stages are integrated with the lexicon editor (see section 5.2.), and the semantic and pragmatic/discourse editor is integrated with both the lexicon editor and the ontology editor (see section 5.3.). The stepped analysis capability allows the developer to see how changes in the analyzers and static knowledge affect the TMRs and supports the semi-automatic production of "gold



(a) (b)

(c)

Figure 2: Static Knowledge Browsing/Editing Interfaces

standard" TMRs for evaluation and other purposes.

5.2. Lexicon Browser/Editor

The Lexicon Browser/Editor allows the knowledge acquirer to look up existing lexical entries, as well as their synonyms and hyponyms, to edit these entries, or to create new ones (Fig. 2 (a)). The editor supports a validation step, insuring that the knowledge entered is both syntactically and semantically correct. The updated information is propagated through the DekadeAPI to OntoSem's database, making available to the analyzer modules. The interface is integrated with the ontology editor, allowing the acquirer to see how the lexical entries relate to their ontological counterparts.

5.3. Ontology Browser/Editor

The Ontology Browser/Editor provides much the same functionality as the lexicon editor, but is targeted at the ontology, an inherently tree-like structure (Fig. 2 (b)). Navigation is done by either navigating a tree view of the ontology, or by keyword lookup. Users can browse, edit, or create ontological entries, and any updates are immediately accessible to the analyzer. The interface facilitates manual acquisition through a variety of ergonomic features, centered around the goal of allowing the user to make as few mouse clicks or type as few characters as possible. The interface also supports validation of the edits made, and is tightly integrated with both the lexicon interface, and the fact repository interface (see section 5.4.).

5.4. Fact Repository Browser/Editor

The Fact Repository Browser/Editor allows the researcher to easily navigate the knowledge automatically created by OntoSem and made persistent in the fact repository, a knowledge base containing remembered instances of ontological concepts and other meaning elements extracted by OntoSem from texts and filtered on the basis of topic relevance. The interface allows manual enhancements to the automatically generated fact repository entries (Fig. 2 (c)). The editor supports ontological validation, and the browser is tightly integrated to the ontology for easy cross-reference between fact repository elements and ontological concepts of which these are instances.

6. Ongoing Work

The DEKADE system is ever changing to fit both new developments in OntoSem, as well as the requests of the developers, knowledge acquirers, and other users. Thus, at the time of writing the following lines of system enhancement are being pursued.

- An interface overhaul is in the works, to support the new model of static knowledge storage and access that relies on the Postgres database system.
- The validation system for each static knowledge acquisition interface is being reworked for improved coverage, efficiency and stability.
- The architecture of the extensible Swing components specific to DEKADE is being cleaned up to make developing new custom panels easier for the researcher.
- Default browser/editor interfaces for some of the auxiliary static knowledge resources of OntoSem are being created (these include such knowledge bases as the onomasticon, a lexicon of proper names).
- Some of the Stepped Analysis interfaces are being retuned for increased usability by the developer: more tight integration with the knowledge browsers, as well as improved editors for the production of "golden" TMRs.
- A fully integrated cross-resource search feature is being developed, which will allow the knowledge acquirer to query the full contents of the static knowledge at the same time.

7. Applications

Since its inception, the DEKADE system (including the DekadeAPI), has been used as a tool for several lines of research, both inside and outside the ILIT lab where DEKADE and OntoSem are currently developed. On the basic science side, the DEKADE system has supported a series of learning experiments, including learning ontological concepts and their places in the ontology through open corpus NLP (the web) [7], as well as learning and validating ontological attribute values through statistical methods over an open corpus (the web) [16].

With respect to practical applications, DEKADE has been successfully used in the SemNews system [10,

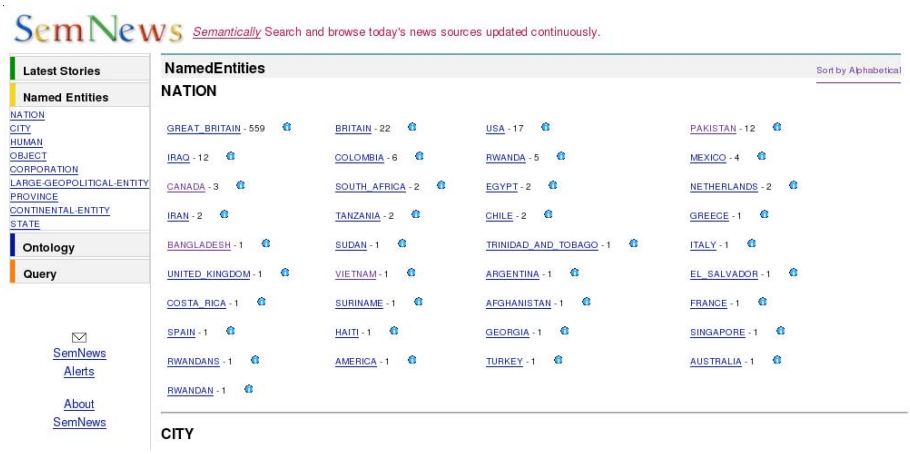


Figure 3: SemNews Interface

11] (Fig. 3), a semantic web annotation project, cataloguing TMR-level annotations of RSS news feeds created by OntoSem. The system uses OntoSem as its backbone NLP system, and OntoSem's static knowledge resources as a default knowledge base.

Another application for which OntoSem provided the basis and DEKADE the environment is EBIDS [18], an NLP-based social engineering email detection system. In EBIDS OntoSem is used through DEKADE to semantically analyze incoming e-mail messages and identify those of them that can be social engineering ("phishing") threats.

8. Evaluation

Evaluating a toolset is a significantly different task compared to evaluating more quantitative research. In the case of a toolset whose goal is to improve the efficiency of developing, testing and operating a system, we can evaluate its performance independently for each of these tasks.

To evaluate its usability to the developer, we can judge whether any usability has been added that was not available (in a practical sense) before, and whether any usability that was previously available has been significantly improved. The DEKADE system does allow for the developer to create gold standard TMRs in a way that is significantly easier than previous methods, as well, the system helps to expedite the process of testing and debugging by allowing the user to step through the main processes of the analysis, and adjust the interim outputs as needed.

To evaluate the benefit to the knowledge acquirer, we can test to see that time and effort is being saved on development. It is evident that this is the case with DEKADE: the knowledge acquirer has an array of tools that allow for efficient browsing, editing and validation. Each of the major static knowledge resources is available to the acquirer through intuitive

interfaces, which are integrated together to improve the overall usage of the system. An acquirer can quickly reference existing knowledge, as well as have any changes they make validated both syntactically and semantically.

To evaluate the usefulness to the researcher, we can look for any added benefit in connectivity that did not exist before. Prior to the inception of the DEKADE system, performing research using the OntoSem system involved having an intimate knowledge of its processors, and static knowledge resources; the DEKADE system allows a researcher to use OntoSem easily and efficiently as a tool, without the burden of learning its software. As shown in section 7 above, this benefit has been realized in several research projects to date.

It is clear from the above discussion that we did not carry out extensive formal user studies to measure efficiency improvements in various tasks when DEKADE was used. Indeed, no funding was so far made available for that purpose in our project. The utility of the tool has been demonstrated simply by its eager adoption by its intended users.

9. Conclusion

In this paper, we have motivated the need to create a full-featured toolset to support an NLP system, by describing the needs of the three sets of users of such a system: the system developer, static knowledge acquirer, and researcher. We have presented the DEKADE system, an integrated toolset solution for this need in the framework of the OntoSem natural language processor. We have described the various functionalities of the DEKADE system, and its supported interfaces and briefly mentioned several outside applications of DEKADE and OntoSem.

10. References

[1] Baker, C., C. Fillmore, J. Low. The Berkeley FrameNet Project. In *Proceedings of COLING-ACL*. Montreal, Canada. 1998.

[2] Baker, C., H. Sato. The FrameNet Data and Software. *Poster and Demonstration at Association for Computational Linguistics*, Sapporo, Japan. 2003.

[3] Beale, S., S. Nirenburg, K. Mahesh. Semantic Analysis in the Mikrokosmos Machine Translation Project. In *Proceedings of the 2nd Symposium on Natural Language Processing*, pp. 297-307, 1995.

[4] Beale, S., B. Lavoie, M. McShane, S. Nirenburg, T. Korelsky. Question Answering Using Ontological Semantics. In *Proceedings of ACL-2004 Workshop on Text Meaning and Interpretation*. Barcelona, Spain. 2004.

[5] Bontcheva, K., V. Tablan, D. Maynard, H. Cunningham. Evolving GATE to Meet New Challenges in Language Engineering. *Natural Language Learning 10 (3/4).* (pp. 349-373). 2004.

[6] Cunningham, H., D. Maynard, K. Bontcheva, V. Tablan. GATE: An Architecture for Development of Robust HLT Applications. In *Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics (ACL02)*. Philadelphia, PA. 2002.

[7] English, J., S. Nirenburg. *Ontology Learning from Text Using Automatic Ontolgical-Semantic Text Annotation and the Web as the Corpus.* Proceedings of the AAAI 2007 Spring Symposium Series on Machine Reading, March 2007.

[8] Fikes, R., J. Jenkins, G. Frank. *JTP: A System Architecture and Component Library for Hybrid Reasoning*. Technical Report KSL-03-01, Knowledge Systems Laboratory, Stanford University, Stanford, CA, USA, 2003.

[9] Gennar, J., et al. The Evolution of Protege: An Environment for Knowledge-Based Systems Development. International Journal of Human-Computer Studies, Volume 58, Issue 1. pp. 89-123. January, 2003.

[10] Java, A., et al. SemNews: A Semantic News Framework. In Proceedings of the Twenty-First National Conference on Artificial Intelligence (AAAI-06). 2006.

[11] Java, A., et al. Using a Natural Language Understanding System to Generate Semantic Web Content. Submitted to the International Journal on Semantic Web and Information Systems (IJSWIS).

[12] Liu, H., P. Singh. ConceptNet: A Practical Commonsense Reasoning Toolkit. *BT Technology Journal, To Appear*. Volume 22, forthcoming issue. Kluwer Academic Publishers. 2004.

[13] Maed, K., et al. The Annotation Graph Toolkit. In *Proceedings of the 1st International Conference on Human Language Technology Research*. pp. 1-6. San Diego, California. 2006.

[14] Miller, G., R. Beckwith, C. Fellbaum, D. Gross, K. Miller. Introduction to WordNet: An On-Line Lexical Database. *International Journal of Lexicography.* (pp. 235-244). 1990.

[15] Nirenburg, S., V. Raskin. Ontological Semantics. SERIES: *Language, Speech, and Communication*, MIT Press, 2004.

[16] Nirenburg, S., D. Dimitroff, J. English, C. Pfeifer. *Three Experiments on Mining the Web for Ontology and Lexicon Learning.* Submitted to the 13th International Conference on Knowledge Discovery and Data Mining (KDD-07).

[17] Singh, P., et al. Open Mind Common Sense: Knowledge acquisition from the general public. In Robert Meersman & Zahir Tari (Eds.), *Lecture Notes in Computer Science: Vol. 2519. On the Move to Meaningful Internet Systems 2002: DOA/CoopIS/ODBASE 2002* (pp. 1223-1237). Heidelberg: Springer-Verlag. 2002.

[18] Stone, A. EBIDS-SENLP: A System to Detect Social Engineering Email Using Natural Language Processing. Unpublished Master's Thesis, University of Maryland Baltimore County. 2007.

[19] Witten, I., E. Frank. Data Mining: Practical machine learning tools and techniques. *2nd Edition, Morgan Kaufmann*. San Francisco. 2005.

[20] http://www.ittvis.com/idl/