

# ‘NAIL’: Artificial Intelligence Software for Learning Natural Language

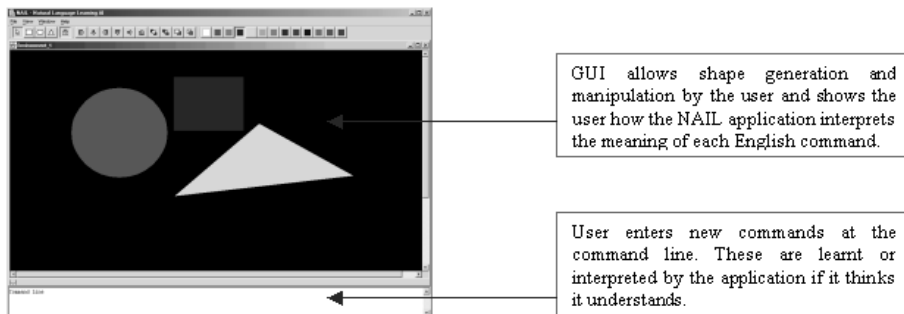
Sam Lievesley<sup>1</sup> and Eric Atwell<sup>1</sup>

School of Computing, University of Leeds, Leeds LS2 9JT, England  
msczsl@comp.leeds.ac.uk, eric@comp.leeds.ac.uk  
<http://www.comp.leeds.ac.uk/oop/nail>

**Abstract.** This paper describes NAIL, a software system to learn the semantics of language by relating written English commands to actions and properties in a simple 2D ‘blocks world’ virtual environment.

## 1 Introduction

Most Computational Natural Language Learning research on learning semantics (eg [1],[2],[3],[4],[5]) assumes that meanings and meaningful relationships can be inferred from text samples alone. We believe that semantics must be learnt as a mapping between language and an external world. This project aims to demonstrate this principle, on a limited scale: NAIL learns mappings from English words, phrases and sentences to meanings in terms of a simple 2D “blocks world”. The software learns about its virtual environment under supervision from a human tutor. The human tutor teaches the computer the meaning of simple language through the use of two interfaces, illustrated in Figure 1.



**Fig. 1.** Graphical User Interface and Command Line for NAIL

A command line terminal is used by the tutor to enter commands written in English. These commands require the program to manipulate or process information from the 2D virtual environment. The 2D virtual environment is displayed

in a graphical user interface (GUI). The human tutor uses this to teach the application the meaning of written commands. Teaching is through learning by example; the tutor directly manipulates the environment using the GUI, showing the program what response is required. The GUI also shows the tutor if the program has manipulated the environment correctly in response to a particular command. The two dimensional virtual environment consists of simple shapes that can be created, deleted, moved and selected. The environment is dynamic as the numbers and types of shape may change throughout a session.

## 2 The NAIL Application

The NAIL application, developed in Java v1.4, was inspired by Winograd's SHRDLU [7]. Unlike SHRDLU the NAIL application does not respond to the user using English, it simply carries out the actions it understands and display the results in the graphical user interface. The application learns new commands by classifying training examples using the WEKA [6],[8] classification tools for machine learning. This will enable further analysis of classification algorithms using any of the WEKA classifiers. The final application will incorporate the most successful classifier from the WEKA package within the program providing real time classification and learning.

NAIL divides the task of learning the meaning of commands between a number of simplified training scenarios. Each scenario trains the system to understand the meaning of a number of simple commands or statements relating to objects in the virtual environment. The training scenarios constrain both the commands that can be entered and the content of the virtual environment. This approach speeds up and simplifies the machine learning process.

### 2.1 Training Scenario 1: Single Object Properties

The purpose of this training scenario is to learn simple adjectives relating to single objects. Each virtual environment contains a single object or shape. The training set used for classification consists of a number of these environments along with binary classification information. The binary classification information is in the form of an adjective and its negative in order to divide the training examples into two classes.

### 2.2 Training Scenario 2: Relationship between Two Objects

The purpose of this training scenario is to learn adjectives that compare the properties or spatial relationships between two world objects.

### 2.3 Training Scenario 3: Relationship between Three Objects

This expands upon training scenario 2 allowing more complicated concepts to be classified. Adjectives such as 'closest' require the comparison of two sets of object property inter-relationships.

## 2.4 Training Scenario 4: Actions That Modify Object Properties

The purpose of this training scenario is to learn the meaning of verbs that modify the properties of single objects. This scenario is different to the previous examples in that a before and after snapshot of object properties must be analysed.

## 3 Future Developments: Understanding Complex Commands

After running a number of training scenarios the software should be able to apply what has been learnt to more complicated command sentences. These sentences will consist of a number of words from different training scenarios that will require the software to select and manipulate certain objects in the virtual environment. This will be achieved by using a number of pre-programmed grammars that will convert sentences into ordered processing stages. These processing stages will select and manipulate objects that are classified by a particular sentence, outputting a final selection or modification to the objects in the virtual environment.

## References

1. Demetriou, G., Atwell, E.: A domain-independent semantic tagger for the study of meaning associations in English text. In: Bunt, H., van der Sluis, I., Thijsse, E. (eds.): Proceedings of IWCS-4: Fourth International Workshop on Computational Semantics. KUB, Tilburg (2001) 67–80
2. Demetriou, G., Atwell, E., Souter, C.: Using lexical semantic knowledge from machine readable dictionaries for domain independent language modelling. In Gavriliadou, M., Carayannis, G., Markantonatou, S., Piperidis, S., Stainhaouer, G. (eds.) Proceedings of LREC2000: Second International Conference on Language Resources and Evaluation Vol.2. ELRA, Athens (2000) 777–782
3. Escudero, G., Mrquez, L., Rigau, G.: A Comparison between Supervised Learning Algorithms for Word Sense Disambiguation. In: Proceedings of CoNLL-2000: Computational Natural Language Learning. Lisbon (2000)
4. Hartrumpf, S.: Coreference Resolution with Syntactico-Semantic Rules and Corpus Statistics In: Proceedings of CoNLL-2001: Computational Natural Language Learning. Toulouse (2001)
5. Sbillot, P., Bouillon, P., Fabre, C.: Inductive Logic Programming for Corpus-Based Acquisition of Semantic Lexicons In: Proceedings of CoNLL-2000: Computational Natural Language Learning. Lisbon (2000)
6. Waikato Environment for Knowledge Analysis (WEKA), <http://www.cs.waikato.ac.nz/ml/weka>, last accessed 26/06/2002.
7. Winograd, T.: Understanding Natural Language. Edinburgh University Press (1972)
8. Witten, I., Eibe Frank, E.: Data Mining: Practical machine learning tools and techniques with Java implementations. Morgan Kaufmann (2000)