Issues in Learning Language in Logic

James Cussens

Department of Computer Science, University of York
Heslington, York, Y010 5DD, UK
jc@cs.york.ac.uk

Abstract. Selected issues concerning the use of logical representations in machine learning of natural language are discussed. It is argued that the flexibility and expressivity of logical representations are particularly useful in more complex natural language learning tasks. A number of inductive logic programming (ILP) techniques for natural language are analysed including the CHILL system, abduction and the incorporation of linguistic knowledge, including active learning. Hybrid approaches integrating ILP with manual development environments and probabilistic techniques are advocated.

1 Introduction

The statistical natural language processing revolution has brought empirical methods of producing natural language resources to the fore. However, despite the long-established use of logic in NLP, most work in natural language learning (NLL) does not take place within a logical framework. Partly this is because many of the successful techniques (e.g. n-gram language models) derive from speech processing where logical approaches are not generally used. Also a logical representation is often seen as unnecessarily complex for NLL. Simpler representations, engineered for specific NLL tasks, are more common.

This chapter will argue that learning language in logic—using a logical representation for NLL—is both practical and desirable for a range of NLL problems. Our argument will fall into two parts. In Section 2 there is a high-level discussion of the role of logical representations in language learning where there are few specific examples. Section 3 in contrast, is much more detailed, examining particularly important issues which arise in actually existing LLL applications. We finish with some tentative predictions in Section 4.

2 Logical Representation in Natural Language Learning

Most logical learning can be defined as inductive logic programming (ILP), so for the sake of completeness, we begin by stating a highly simplified version of the ILP task in Table 1. The flexibility and expressivity of logic are amongst the prime reasons for using ILP. This flexibility and expressivity has also led to the development of a variety of logic-based resources for natural language
Table 1. Highly simplified version of the ILP problem

\textbf{Given} background knowledge \(B\) and data \(E\)

\textbf{Find} a hypothesis \(H\) such that

\[ B \land H \models E \]

where \(B\), \(H\) and \(E\) are all logic programs.

processing (NLP). If we wish to \textit{learn} such resources then this section argues that it is appropriate that the learning process also stays within this ‘native’ logical representation. I argue here that moving to a less expressive learning framework such as attribute-value learning will mean that some flexibility or expressivity will be lost.

In cases where the output of learning is required to be logical it is easy to motivate logical learning. Learning problems of this type include learning definite clause grammars [11, 21], semantic grammars [27], and learning rules to translate between logically represented semantic forms from different languages [5].

In other cases the form of the data motivates a logical representation. For example, text can be annotated with information such as part-of-speech (PoS) tags, syntactic parse trees or semantic interpretations. As we move from ‘lower-level’ information such as PoS tags to ‘higher-level’ information such as semantic annotation, the case for a logical representation becomes stronger. This is simply because logic has been designed to elegantly represent complex and structured information. A good example is the use of \textit{quasi-logical forms (QLFs)} to provide semantic interpretation. Table 2 shows a pair of QLFs representing (at a semantic level) the English sentence \textit{List the prices} and the French sentence \textit{Indiquez les tarifs}. For details of QLF syntax and semantics see [3]. This sort of data is used by Boström [5] to learn transfer rules between French and English QLFs. It is difficult to see how the complex terms required to represent this sort of information could be adequately translated to a non-logical representation.

Sometimes the learning problem is such that a logical representation is not \textit{essential} but is more convenient than simpler approaches. For example, when learning from unannotated text it is possible to represent sequences of words (\(n\)-grams) by feature vectors of length \(n\). The same holds true of text annotated with information such as PoS tags. This is because, in practice, all such sequences will be of bounded length. If the longest sequence in the data were of length \(n = 20\), then the sentence \textit{James loves Gill} would be represented as the feature vector

\[(W1 = \text{James}, W2 = \text{loves}, W3 = \text{Gill}, W4 = \emptyset, \ldots, W20 = \emptyset)\]

where \(\emptyset\) is a null value. However, this ‘flat’ representation is unwieldy since all sequences except those of maximal length will involve superfluous null values. A much more compact representation would be to use lists, which are first-order terms, so that \textit{James loves Gill} becomes

\[ ['\text{James}', '\text{loves}', '\text{Gill'}] \]