

A machine-learning approach to the identification of WH gaps

Derrick Higgins

Educational Testing Service

dchiggin@alumni.uchicago.edu

Abstract

In this paper, we pursue a modular, statistical approach to WH dependencies, using a feedforward network as our modeling tool. The empirical basis of this model and the availability of performance measures for our system address deficiencies in earlier computational work on WH gaps, which require richer sources of semantic and lexical information in order to run. The statistical nature of our models allows them to be simply combined with other modules of grammar, such as a syntactic parser.

1 Overview

This paper concerns the phenomenon of *WH dependencies*, a subclass of *unbounded dependencies* (also known as *\bar{A} dependencies* or *filler-gap structures*). WH dependencies are structures in which a constituent headed by a WH word (such as “who” or “where”) is found somewhere other than where it belongs for semantic interpretation and subcategorization.

\bar{A} dependencies have played an important role in syntactic theory, but discovering the location of a gap corresponding to a WH phrase found in the syntactic representation of a sentence is also of interest for computational applications. Identification of the syntactic gap may be necessary for interpretation of the sentence, and could contribute to a natural language understanding or machine translation application. Since WH dependencies also tend to distort the surface subcategorization properties of verbs, identifying gaps could also aid

in automatic lexical acquisition techniques. Many other applications are imaginable as well, using the gap location to inform intonation, semantics, collocation frequency, etc.

The contribution of this paper consists in the development of a machine-learning approach to the identification of WH gaps. This approach reduces the lexical prerequisites for this task, while maintaining a high degree of accuracy. In addition, the modular treatment of WH dependencies allows the model to be easily incorporated with many different models of surface syntax. In ongoing work, we are investigating ways in which our model may be combined with a syntactic parser.

The idea that the task of associating a WH element with its gap can be done in an independent module of grammar is not only interesting for reasons of computational efficacy. The fact of unbounded dependencies has played a central role in the development of linguistic theory as well. It has been used as an argument for the necessity of transformations (Chomsky, 1957), and prompted the introduction of powerful mechanisms such as the **SLASH** feature of GPSG (Gazdar et al., 1985). To the extent that these phenomena can be described in an independent module of grammar, our theory of the syntax of natural language can be accordingly simplified.

2 Previous Work

In theoretical linguistics, WH dependencies have typically been dealt with as part of the syntax (but cf. Kuno, Takami & Wu (1999) for an alternative approach). Early generative treatments used a *WH-movement* transformation (McCawley, 1998) to describe the relationship between a WH phrase and its gap, while later work in the Government &

Binding framework subsumes this transformation under the general heading of overt \bar{A} Movement (Huang, 1995; Aoun and Li, 1993). Feature-based syntactic formalisms such as GPSG use feature-percolation mechanisms to transfer information from the location in which a WH phrase is subcategorized for (the gap), to the location where it is realized (the filler position) (Gazdar et al., 1985; Pollard and Sag, 1994).

Most work in computational linguistics has followed these theoretical approaches to WH dependencies very closely. Berwick & Fong (1995) implement a transformational account of WH gaps in their Government & Binding parser, although the grammatical coverage of their system is extremely limited. The SRI Core Language Engine (Alshawi, 1992) incorporates a feature-based account of WH gaps known as “gap-threading”, which is essentially identical to the feature-passing mechanisms used in GPSG. Both systems require an extensive lexical database of valency information in order to identify potential gap locations.

While there are no published results regarding the accuracy of these methods in correctly associating WH phrases and their gaps, we feel that these methods can be improved upon by adopting a corpus-based approach. First, deriving generalizations about the distribution of WH phrases directly from corpus data addresses the problem that the data may not conform to our theoretical preconceptions. Second, we hope to show that much of the work of identifying WH dependencies can be done without access to the subcategorization frame of every verb and preposition in the corpus, which is a prerequisite for the methods mentioned above.

The only previous work we are aware of which addresses the task of identification of WH gaps from a statistical perspective is Collins (1999), which employs a lexicalized PCFG augmented with “gap features”. Unfortunately, our results are not directly comparable to those reported by Collins, because his model of WH dependencies is integrated with a syntactic parser, so that his system is responsible for producing syntactic phrase structure trees as well as gap locations. Since our model takes these trees as given, it identifies the correct gap location more consistently. Integration

of our model of WH dependencies with a parser is a goal of future development.

3 Modeling WH dependencies

The task with which we are concerned in this section is determining the location of a WH gap, given evidence regarding the WH phrase and the syntactic environment. Following much recent work which applies the tools of machine learning to linguistic problems, we will treat this as an example of a classification task. In Section 3.2 below, we describe the neural network classifier which serves as our grammatical module responsible for WH gap identification.

3.1 Data

The data on which the classifiers are trained and tested is an extract of 7915 sentences from the Penn Treebank (Marcus et al., 1993), which are tagged to indicate the location of WH gaps. This selection comprises essentially all of the sentences in the treebank which contain WH gaps. Figure 1 shows a simplified example of WH-fronting from the Penn Treebank in which the WH word *why* is associated with the matrix VP node, despite its fronted position in the syntax. Note that it cannot be associated with the lower VP node. The Penn Treebank already indicates the location of “WH-traces” (and other empty categories), so it was not necessary to edit the data for this project by hand, although they were automatically pre-processed to prune out any nodes which dominate only phonologically empty material.

In treating the identification of WH gaps as a classification task, however, we are immediately faced with the issues of identifying a finite number of *classes* into which our model will divide the data, and determining the features which will be available to the model.

Using the movement metaphor of syntactic theory as our guide, we would ideally like to identify a complete *path* downward from the surface syntactic location of the WH phrase to the location of its associated gap. However, it is hard to see how such a path could be represented as one of a finite number of classes. Therefore, we treat the search downward through the phrase-structure tree for the location of a gap as a Markov process. That is, we

Figure 1: Simplified tree from the Penn Treebank. The fronted WH-word ‘why’ is associated with a gap in the matrix VP, indicated by the empty constituent (-NONE- *T*-2).

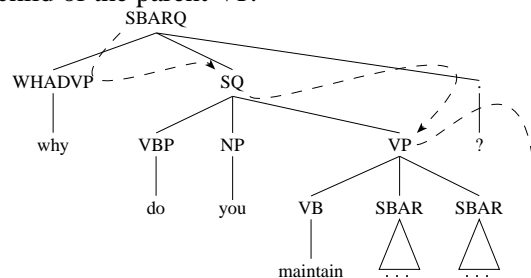
```
(SBARQ
 (WHADVP-2 (WRB Why) )
 (SQ (VBP do)
 (NP-SBJ (PRP you) )
 (VP (VB maintain)
 (SBAR (-NONE- 0)
 (S
 (NP (DT the) (NN plan) )
 (VP (VBZ is)
 (NP
 (DT a)
 (JJ temporary)
 (NN reduction) ))))
 (SBAR
 (WHADVP-1 (WRB when) )
 (S
 (NP (PRP it) )
 (VP (VBZ is) (RB not)
 (NP (-NONE- *?*) )
 (ADVP (-NONE- *T*-1) ))))
 (ADVP (-NONE- *T*-2) ))
 (. ?) )
```

begin at the first branching node dominating the WH operator, and train a classifier to trace downward from there, eventually predicting the location of the gap. At each node we encounter, the classifier chooses either to recurse into one of the child nodes, or to predict the existence of a gap between two of the overt child nodes. (Since the number of daughters in a subtree is limited, the number of classes is also bounded.) This decision is conditioned only on the category labels of the current subtree, the nature of the WH word extracted, and the depth to which we have already proceeded in searching for the gap. This greedy search process is illustrated in Figure 2.

Each sentence was thus represented as a series of records indicating, for each subtree in the path from the WH phrase to the gap, the relevant syntactic attributes of the subtree and WH phrase, and the action to be taken at that stage (e.g., **GAP-0**, **RECURSE-2**).¹ Sample records are shown in Figure 3. The “join category” is defined as the lowest node dominating both the WH phrase and

¹We indicate the target classes for this task as **RECURSE- n** , indicating that the Markov process should recurse into the n^{th} daughter node, or **GAP- n** , indicating that a gap will be posited at offset n in the subtree.

Figure 2: Illustration of the path a classifier must trace in order to identify the location of the gap from Figure 1. At the top level, it must choose to recurse into the SQ node, and at the second level, into the VP node. Finally, within the VP subtree it should predict the location of the gap as the last child of the parent VP.



its associated gap; the meanings of the other features in Figure 3 should be clear.

3.2 Classifier

For our classifier model of WH dependencies, we used a simple feed-forward multi-layer perceptron, with a single hidden layer of 10 nodes. The data to be classified is presented as a vector of features at the input layer, and the output layer has nodes representing the possible classes for the data (**RECURSE-1**, **RECURSE-2**, **GAP-0**, **GAP-1**, etc.). At the input layer, the information from records such as those in Figure 3 is presented as binary-valued inputs; i.e., for each combination of feature type and feature value in a record (say, *mother cat* = *S*), there is a single node at the input layer indicating whether that combination is realized.

We trained the connection weights of the network using the **quickprop** algorithm (Fahlman, 1988) on 4690 example sentences from the training corpus (12000 classification stages), reserving 1562 sentences (4001 classification stages) for validation to avoid over-training the classifier. In Table 1 we present the results of the neural network in classifying our 1663 test sentences after training.

4 Conclusion

These performance levels seem quite good, although at this point there are no published results

Figure 3: Example records corresponding to the sentence shown in Figures 1 & 2

target class: RECURSE-2	target class: RECURSE-3	target class: GAP-3
depth: 0	depth: 1	depth: 2
WH cat: WHADVP	WH cat: WHADVP	WH cat: WHADVP
WH lex: why	WH lex: why	WH lex: why
join cat: SBARQ	join cat: SBARQ	join cat: SBARQ
mother cat: SBARQ	mother cat: SQ	mother cat: VP
daughter cat1: WHADVP	daughter cat1: VBP	daughter cat1: VB
daughter cat2: SQ	daughter cat2: NP	daughter cat2: SBAR
daughter cat3: .	daughter cat3: VP	daughter cat3: SBAR
daughter cat4: UNDEFINED	daughter cat4: UNDEFINED	daughter cat4: UNDEFINED
daughter cat5: UNDEFINED	daughter cat5: UNDEFINED	daughter cat5: UNDEFINED
daughter cat6: UNDEFINED	daughter cat6: UNDEFINED	daughter cat6: UNDEFINED
daughter cat7: UNDEFINED	daughter cat7: UNDEFINED	daughter cat7: UNDEFINED

Table 1: Test-set performance of network

	Percentage Correct
Complete path	1530/1663 = 92.0%
String location	1563/1663 = 94.0%
Each stage	4093/4242 = 96.5%

for other systems to which we can compare them. We take this level of success as an indication of the feasibility of our data-driven, modular approach. Additionally, our approach has the advantage of wide coverage. Since it does not require an extensive lexicon, and is trained on corpus data, it is easily adaptable to many different NLP applications. Also, since the treatment of WH dependencies is factored out from the syntax, it should be possible to employ a simple model of phrase structure, such as a PCFG.

In future work, we hope to explore this possibility, by combining the classifier model of WH dependencies developed here with a syntactic parser, so that our results can be directly compared with those of Collins (1999). The general mechanism for combining these two models is the same one used by Higgins & Sadock (2003) for combining a quantifier scope component with a parser, taking the syntactic component to define a prior probability $P(S)$ over syntactic structures, and the additional component to define the probability $P(K|S)$, where K ranges over the values which the other grammatical component may take on.

References

Hiyan Alshawi, editor. 1992. *The Core Language Engine*. MIT Press.

Joseph Aoun and Yen-hui Audrey Li. 1993. *The Syntax of Scope*. MIT Press, Cambridge, MA.

Robert C. Berwick and Sandiway Fong. 1995. A quarter century of parsing with transformational grammar. In J. Cole, G. Green, and J. Morgan, editors, *Linguistics and Computation*, pages 103–143.

Noam Chomsky. 1957. *Syntactic Structures*. Janua Linguarum. Mouton, The Hague.

Michael Collins. 1999. *Head-Driven Statistical Models for Natural Language Parsing*. Ph.D. thesis, University of Pennsylvania, Philadelphia, PA.

S. E. Fahlman. 1988. An empirical study of learning speed in back-propagation networks. Technical Report CMU-CS-88-162, Carnegie Mellon University.

Gerald Gazdar, Ewan Klein, Geoffrey Pullum, and Ivan Sag. 1985. *Generalized Phrase Structure Grammar*. Harvard University Press, Cambridge, MA.

Derrick Higgins and Jerrold M. Sadock. 2003. A machine learning approach to modeling scope preferences. *Computational Linguistics*, 29(1):73–96.

C.-T. James Huang. 1995. Logical form. In G. Webelhuth, editor, *Government and Binding Theory and the Minimalist Program*, pages 125–175. Blackwell, Oxford.

Susumu Kuno, Ken-Ichi Takami, and Yuru Wu. 1999. Quantifier scope in English, Chinese, and Japanese. *Language*, 75(1):63–111.

M. Marcus, S. Santorini, and M. Marcinkiewicz. 1993. Building a large annotated corpus of English: the Penn Treebank. *Computational Linguistics*, 19(2):313–330.

James D. McCawley. 1998. *The Syntactic Phenomena of English*. University of Chicago Press, Chicago, second edition.

Carl Pollard and Ivan Sag. 1994. *Head-Driven Phrase Structure Grammar*. University of Chicago Press, Chicago.