1 Discussion

The previous chapters have highlighted several systems for classifying texts as native and nonnative based on grammatical features. The accuracy of these systems varies from approximately 70% to 90%. While such classifiers are interesting, and may have some utility in and of themselves, it is worth exploring whether these can form the foundation of an educational tool for learners of English. An effort has been made, when examining the decision trees of the various classification systems, to find linguistic reasons behind the classifiers' choice of attributes. It has been found that some attributes are used in ways which are very easy to interpret, based on the grammars of English and Spanish and the concept of L1-transfer, and others seem quite incomprehensible. This linguistic analysis is important, as it gives clues as to which attributes would play a useful part of a learning tool, and which should be excluded. This chapter seeks to further analyze the attributes used by the classifiers in light of their utility in a learner's tool, and to sketch out a possible design of such a tool.

1.1 Learner's Tool Design

Much of the complexity of any piece of user-oriented software is in the interface, but that will not be discussed in much detail here. Suffice it to say that the program would allow the user, a learner of English, to input a selection of English text written by that user. Because this system would be based on the frequency of textual attributes, the inputted text would need to be of a sufficient length such that the system could measure these frequencies with some level of statistical significance. Having inputted a text, the user would then be presented with the output. This output would either inform the user that certain features had led the system to consider the text to be identifiably nonnative, or that the system was unable to distinguish it from native texts. In the former case, the user would be shown which features of the text were responsible for this classification, and, when possible, would be

shown all passages of the text that exhibited these features.

Identifying these features could be easily done if the system is using decision trees, whether generated by the C4.5 algorithm or by a similar system. Consider any of the decision trees shown in the previous chapters. When one of these trees identifies a text, presented to it as a set of attribute and value pairs, as being nonnative, it has associated that text with a particular leaf on the tree. From this node to the root there is a unique path which passes through all of the decision nodes which were used in classifying the text. Each of these decision nodes considers a single, though not necessarily unique, attribute, and these attributes together make up the set of attributes that most strongly mark the text as being nonnative. It would not be difficult, using Weka for instance, to implement a version of C4.5 that generates decision trees which output all attributes used in a particular classification, in addition to the class itself. The system could use a single tree, trained on all attributes, or it could use multiple trees. These trees might correspond to grammatically distinct attribute sets, as do the trees shown in the previous chapters, or they might be the various trees of a Random Forest classifier. Using multiple trees may enable the system to present the user with a wider range of features.

The passages which exhibit the telltale features could then be located using tables which had been generated when the system initially identified the features. Many of these textual features would be examples of overuse of various grammatical constructions, and the user could focus on the outputted passages as he or she revises the text. Others, however, would be examples of underuse. In this case, the system would only be able to identify positive examples in the text, or perhaps no examples at all, and it would be up to the user to find the appropriate places in which to use the relevant construction. A more sophisticated system might group certain features together in complementary or nearly complementary pairs (e.g. modals and phrasal modals), and then use the complement of the underused feature to suggest areas for revision. The user would then edit the text and have the system reevaluate it. Eventually, if the user is able to make appropriate changes, there would be a change in

how the text is classified. The system may still classify it as nonnative, but would do so at a different leaf node in the tree, with a different set of responsible attributes. The cycle of revision and reevaluation could continue until the system classifies the text as native.