6 Conclusions

The thesis that I have set out to defend in this dissertation has been that semantic relationships in sentences form a model of a text and that it is possible to acquire this model interactively. There are a number of ways to make the user-assisted acquisition relatively painless: define general, domain-independent semantic relationships; find clues in syntax that hint at underlying semantic relationships; ask the user simple, well-directed questions when consultation is necessary; apply simple learning techniques to approved analyses to allow the system to act more autonomously on future sentences.

6.1 Summary

The bulk of this dissertation was about the design and implementation of parts of TANKA's semantic analyzer. The purpose of HAIKU is to recognize semantic relationships in the sentences of a text. Chapters 2, 3 and 4 dealt with HAIKU's three semantic analysis modules.

6.1.1 Clause Level Relationship Analysis

In chapter 2 I described the clause level relationship analyzer. CLRs are assigned to clauses in coordinate, subordinate and correlative syntactic relationships. The CLR marker dictionary lists a subset of candidate CLRs appropriate to the clausal connective in a sentence. HAIKU holds competitions between the candidates using preference rules that choose between pairs of CLRs. The rules prefer one CLR over another depending on the connective polarity and the verb phrase polarity, tense and modality. The CLR with the most points at the end of all competitions is presented to the user for approval. To help with future assignments, approved CLRs and clausal attributes are stored and generalized on unequal attribute values.

6.1.2 Case Relationships

Since case analysis was previously described by Delisle (1994), in chapter 3 I showed in detail the process of choosing a set of semantic relationships. Initial selection comes from a broad review of existing systems in theoretical and computational linguistics, seeding the set with linguistically and practically motivated relationships. Building a marker dictionary grounds the set in the surface-syntactic phenomena that express the relationships. Marker dictionary construction also exposes weaknesses in the coverage of the set.

In chapter 3 I also described a methodical evaluation of the cases using their distribution in texts and among the case markers. The evaluation criteria were *generality*, *completeness* and *uniqueness* (absence of *redundant* and *superfluous* cases).

6.1.3 Noun Modifier Relationship Analysis

The noun modifier relationship analyzer described in chapter 4 assigns NMRs to each of the modifiers of a noun. Premodifiers must be bracketed into local modifier-head pairs, each of which received its own NMR. The semi-automatic bracketer handles adjective and noun premodifier sequences of any length. It learns from reduced subbracketings of previous phrases to decide whether a subphrase is left-branching or right-branching.

HAIKU assigns NMRs to modifier-head pairs from bracketed premodifier sequences and from postmodifying prepositional phrases and appositives. Modifier-head-marker triples are compared to previous triples. NMRs associated with the most similar such previous triples are candidates for the relationship between the modifier and head under analysis. One of these candidates is chosen as the best NMR based on its frequency of occurrence in previous assignments. The NMR analyzer also generates automatically taxonomic relationships between a compound and its head.

6.2 Goals Revisited

In chapter 1 I presented five goals for the project. I will now return to those goals and relate them to the results presented in the preceding chapters.

6.2.1 Semantic Relationships

The first goal was to construct usable, general purpose sets of semantic relationships. I have described the construction of three such sets: clause level relationships, cases and noun modifier relationships. Each of the lists was inspired by a union of semantic relationships found in research in discourse analysis, in case theory and valency theory and on noun phrases.

The clause level relationships cover the senses of a large number of coordinators and subordinators found in conventional and electronic dictionaries. In the *building code*, *clouds* and *small engines* experiments no new CLRs were needed to account for the semantic relationships between events represented by clauses in those texts.

HAIKU's case marker dictionary contains several hundred lexical items (atomic and phrasal markers) whose many senses are covered by the cases. Some cases are marked by fewer markers than others, but no case exists for only a single marker. Inspection of the representation of each case in the marker dictionary identified potentially over-general or over-specific cases.

The existing case set has also been sufficiently complete to account for verb argument relationships in many texts, including (most recently) the *clouds* text and the *small engines* text. On the other hand, several cases were not assigned in those experiments. In chapter 3 I rejected the possibility that cases such as Recipient, TimeFrom and TimeTo are superfluous, even if they appear to occur rarely in the test texts. The inclusion of similar cases in many other case lists is indirect evidence of their utility.

Semantic relationships within noun phrases in the *sparc* and *small engines* texts were all covered by the existing NMRs. Nonetheless, there are possibilities for new relationships (see section 5.3.2). Experiments with more texts will determine if the coverage of the current set of NMRs is sufficient.

6.2.2 Semi-Automatic Recognition of Semantic Relationships

The second goal from chapter 1 was the completion of the HAIKU semantic analyzer. Chapters 2, 3 and 4 contain details of the design and implementation of each of its modules. As planned, the components make use of linguistic knowledge and of small amounts of closed, domain-independent knowledge. They learn from user assignments to make better suggestions to the user. The success of the implementation is supported by the data from experiments, as summarized in the following sections.

6.2.3 Learning

The third goal was to evaluate HAIKU's ability to learn from previous analyses to make better suggestions to the user. Results of the experiments described in chapters 2, 3 and 4 show that as more text is analyzed, the proportion of correct analyses by the system increases steadily.

For the *clouds* and *small engines* texts, the clause level relationship analyzer got 76% of CLR assignments correct using preference rules only. The problem with the preference rules is that if they prefer the wrong CLR for a given input, they will always prefer the wrong CLR for similar inputs. Experience using HAIKU has revealed that correcting the system on the same mistakes is simple, but annoying.

Learning from user assignments helps HAIKU avoid such behaviour, but there is the problem of knowing when to use stored CLR attribute patterns and when to stay with the preference rules. For the 127 CLR assignments from the *clouds* and *small engines* texts, CLRA got 77% correct using stored attribute patterns and preference rules. More experiments are needed to determine if there is a significant difference in performance between the two methods and also to see if performance improves over time when using stored attribute patterns.

HAIKU assigned 439 case patterns in the *clouds* experiment and 584 patterns in the *small engines* experiment. For both experiments the proportion of clauses for which the correct CP was among the system's suggestions increased over the course of the experiment. For the *clouds* experiment HAIKU had correctly assigned 50% of the CPs after 208 clauses. It correctly assigned 72% of the CPs for the rest of the text. In the *small engines* experiment HAIKU had correctly assigned half of the CPs after the first 100 clauses. For the rest of the clauses, it correctly assigned 68%. Figure 4 and Figure 5 in chapter 3 illustrate how the cumulative number of correct assignments by HAIKU increased at a higher rate than user assignments throughout both experiments.

In the *sparc* experiment HAIKU's bracketer made 63% of the 194 bracketing decisions correctly: 52% in the first half of the experiment, 74% in the second half. For the *small engines* experiment the bracketer made 62% of the 164 decisions correctly. For the first quarter of that experiment the number hovered just over 50% after which it jumped fairly quickly up to about 65% where it stayed for the rest of the experiment. Always guessing left branching would have resulted in 91% success for the *small engines* text, but only 45% for the *sparc* text. Based on others' observations of branching frequencies, it is unlikely that either of these two texts is representative of texts in general. Yet the *sparc* text shows that the predominance of left-branching compounds is not universal.

The noun modifier relationship analyzer assigned 72% of the NMRs in the *sparc* experiment correctly: 51% in the first third of the experiment, 83% for the rest of the text. The *small engines* experiment showed similar numbers: 49% of the first 100 assignments were made correctly by the system, 71% of the assignments for the rest of the text. On average NMRA made 69% of the assignments correctly over the entire text. The improvements show that NMRA can learn from user assignments. Furthermore, since only 57% of the *sparc* assignments and 43% of the *small engines* assignments were perfect matches on reduced modifier-head pairs, partial matching was responsible for a significant portion (15% and 26% respectively) of the correct analyses.

6.2.4 Coverage

The fourth goal was to evaluate HAIKU's coverage of semantic relationships in a text and its sensitivity to errors in parse trees and missing parse trees. The system can recover many semantic relationships even from incorrect or fragmentary parse trees.

It was no surprise to discover that clause level relationship analysis suffered the most from parse errors in the experiments. HAIKU got the opportunity to assign CLRs for 67% of the clause-clause constructions in the *clouds* text. In the more syntactically complex *small engines* text, only 28% of constructions requiring CLRs were analyzed.

Case analysis is the least sensitive to parse errors because it allows the user to correct mistakes in case marker patterns. Sampling 100 verbs at random from the *small engines* text showed that HAIKU extracted between 92% and 99% of the case patterns in the text (at 95% confidence). The number would have been much smaller if the system were limited to assigning case patterns to correct CMPs only. In the *clouds* experiment, 69% of the CMPs constructed from the parse trees were correct. For the *small engines* experiment, only 55% of CMPs were correct.

Noun modifier relationship analysis works with the smallest parse tree fragments and should be the least sensitive to parse errors. With DIPETT's partial parsing facility, many noun phrases are parsed correctly even if the parser cannot connect their parses to larger trees. Unlike in case analysis, however, the user is not given the opportunity to repair incorrect parses. Sampling 100 modifier-noun pairs in the *small engines* text showed that between 79% and 92% of the pairs in the text received NMRs (at 95% confidence).

6.2.5 User Burden

The final goal stated in chapter 1 was to evaluate the burden that semi-automatic HAIKU analysis places on the user. Experiments have shown that the system can be used to recover multiple semantic relationships from several hundred sentences in just a few days (starting from scratch) without overly taxing the user.

Burden is evaluated in terms of the onus ratings (see section 1.4.4) assigned to each interaction during the *clouds* and *small engines* experiments. The onus ratings assigned for interactions in both experiments appear in Table 37. In those experiments 1739 of the

		onus 0	1	2	3	average
clouds	*CLRA*	50	1	0	0	0.02
	CA	384	50	5	0	0.14
small engines	*CLRA*	17	4	0	0	0.19
	CA	480	89	15	0	0.20
	NMRA	808	71	7	0	0.10

Table 37: Onus ratings for the clouds *and* small engines *experiments*

1981 HAIKU interactions (88%) were considered trivial, while only 27 of the interactions (1%) were considered difficult.

Burden can also be evaluated by looking at the number of HAIKU interactions per sentence and the amount of time spent on each sentence. In the *clouds* experiment there were on average 0.10 CLRs and 0.86 case patterns assigned in each sentence, requiring on average 1 minute and 49 seconds. In the *small engines* experiment there were 0.04 CLRs, 1.05 case patterns and 1.59 NMRs per sentence, coincidentally also requiring 1 minute 49 seconds. In both experiments the average time per sentence relative to the number of HAIKU interactions decreased over the course of the experiment.

6.3 Closing Words

Semantic relationships in sentences form a model of a text that can be acquired interactively, with relatively little burden on the user. This thesis is defended through the satisfaction of five goals: the construction of sets of general, portable semantic relationships; the implementation of a semi-automatic system that recognizes the semantic relationships in sentences using syntactic and linguistic clues; confirmation of the system's ability to learn to work more autonomously; confirmation that the semantic relationships are sufficient to model complete texts; and confirmation that the burden placed on the user is low. The design, implementation and evaluation of HAIKU directly addressed each of the goals.

The conclusion is that the interactive recovery of semantic relationships in a text is viable. The syntax does hold clues about the underlying semantic relationships between syntactic elements in a sentence. HAIKU can use these clues to recognize a good portion of the semantic relationships in a text automatically, By learning from mistakes corrected by a user it can improve its performance. Finally, there are ways to elicit knowledge from the user without requiring open-ended subjective knowledge entry.

Systems that acquire deep semantic knowledge from texts fully automatically are not forthcoming. This dissertation has shown that it is feasible to acquire some knowledge from texts starting from scratch—without big semantic knowledge bases, without big tagged corpora, with just a little help from a user and a little learning.