MITSUBISHI ELECTRIC RESEARCH LABORATORIES
http://www.merl.com

The Evaluation of Anapron: A Case Study in Evaluating a Case-based System

Andrew R. Golding, Paul S. Rosenbloom

TR94-05 December 1994

Abstract

This paper presents a case study in evaluating a case-based system. It describes the evaluation of Anapron, a system that pronounces names by a combination of rule-based and case-based reasoning. Three sets of experiments were run on Anapron: a set of exploratory measurements to profile the systemś operation; a comparison between Anapron and other name-pronunciation systems; and a set of studies that modified various parts of the system to isolate the contribution of each. Lessons learned from these experiments for CBR evaluation methodology and for CBR theory are discussed.

Working Notes of the AAAI-94 Workshop on Case-Based Reasoning, Seattle, WA, 1994, pages 84-90

1. First printing, TR94-05, May 1994

The evaluation of Anapron: A case study in evaluating a case-based system[1]

Andrew R. Golding
Mitsubishi Electric Research Labs
201 Broadway, 8th Floor
Cambridge, MA 02139
golding@merl.com

Paul S. Rosenbloom
ISI and Computer Science Department
University of Southern California
4676 Admiralty Way
Marina del Rey, CA 90292
rosenbloom@isi.edu

Abstract

This paper presents a case study in evaluating a case-based system. It describes the evaluation of Anapron, a system that pronounces names by a combination of rule-based and case-based reasoning. Three sets of experiments were run on Anapron: a set of exploratory measurements to profile the system's operation; a comparison between Anapron and other name-pronunciation systems; and a set of studies that modified various parts of the system to isolate the contribution of each. Lessons learned from these experiments for CBR evaluation methodology and for CBR theory are discussed.

1 Introduction

This paper describes the evaluation of Anapron as a case study in evaluating a case-based system. Anapron works in the domain of name pronunciation, and is based on a general architecture for combining rule-based and case-based reasoning. The central hypothesis embodied by the system is that combining rules and cases allows it to achieve higher accuracy than it could with either knowledge source alone. The intuition for this is that rules and cases have complementary strengths: rules capture broad trends in the domain, while cases are good at filling in small pockets of exceptions in the rules.

The central hypothesis and others about the system were tested in a three-part evaluation: an initial set of measurements to profile the system's operation and detect any abnormalities; a comparison between Anapron and other name-pronunciation systems; and a set of studies that systematically modified various components of the system to see how much each was contributing to overall performance. This third part provided a key result — that both rules and cases were needed for the system to achieve its best accuracy. This confirmed the central hypothesis that combining rules and cases allows the system to achieve higher accuracy than it could have gotten with either one alone.

The rest of this paper is organized as follows: the next section briefly describes Anapron, as needed for understanding the experiments. The subsequent section describes the experiments themselves. Finally, lessons learned from the experiments both for CBR evaluation methodology and for CBR theory are discussed.

2 Anapron

Anapron is a name-pronunciation system based on a general method for combining rule-based and case-based reasoning. The sections below describe the general method, the task of name pronunciation, and the application of the method to this task. For more on the method, see Golding and Rosenbloom (1991). A thorough treatment of both the method and its application to name pronunciation can be found in Golding (1991).

2.1 Combining RBR and CBR

The central idea of the method for combining RBR and CBR is to apply the rules to a target problem to obtain a first approximation to the answer, and to draw analogies from cases to cover exceptions to the rules. Using cases in this way tends to be easier than fine-tuning the rules to cover all contingencies. The method is designed for domains where a reasonable set of rules is already available, and thus it makes sense to take the rules as a starting point, rather than applying CBR from scratch.

The central idea of the method is expressed in the RC-Hybrid procedure of Figure 1. The procedure treats problem solving as a process of applying operators to the target problem until it is solved.

[1]This research was sponsored by NASA under cooperative agreement number NCC 2–538, and by a Bell Laboratories PhD fellowship to the first author. Computer facilities were partially provided by NIH grant LM05208. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of NASA, the US Government, Bell Laboratories, or the National Institute of Health.

Procedure RC-hybrid(*Problem*)

Until *Problem* is solved **do**:

(a) RBR: Use the rules to select an operator.
(b) CBR: Look for analogies that contradict the operator suggested by RBR.
(c) Combination: Decide between the operators suggested by RBR and CBR.

Figure 1: Top-level procedure for combining rule-based and case-based reasoning.

The procedure applies one operator on each iteration. It chooses the operator in three steps. In the RBR step, it selects an operator to apply via the rules. In the CBR step, it looks for analogies suggesting operators that contradict the one suggested by RBR. In the combination step, it decides which operator to actually apply — the one suggested by RBR or the one suggested by CBR.

To decide between RBR and CBR in this last step, the procedure evaluates the analogy proposed by CBR. It chooses the CBR operator if and only if this analogy is found to be *compelling*. Compellingness is based partly on the similarity score for the analogy — this is the degree of similarity between the analogical source and target as given by the similarity metric. It is also based on an *empirical verification* of the analogy. Empirical verification entails extracting the generalization behind the analogy and testing it out on other examples in the case library. There are two results: accuracy, which is the proportion of examples for which the generalization was found to be correct; and significance, which is 1 minus the probability of getting that high an accuracy merely by chance. The analogy is then said to be compelling iff its similarity score, accuracy, and significance satisfy the Compelling-p predicate of Figure 2. The values SS_0, SS_+, A_0, and S_0 in the predicate definition are thresholds that are set by a learning procedure that generates training analogies for itself from the case library.

Before the RC-hybrid procedure can be run, it is necessary to create an indexed case library. This is done in two preprocessing steps. The first, *rational reconstruction* (RR), takes as input a set of problem/answer pairs in the domain. For each pair, it infers the (likely) sequence of operators that were applied to the given problem to produce the given answer. The second preprocessing step, *prediction-based indexing* (PBI), then stores each operator inferred by RR as a positive or negative exemplar of the rules, according to whether that operator agrees with the operator predicted by the rules.

Compelling-p($\mathcal{A}$) $\Longleftrightarrow$
similarity-score($\mathcal{A}$) $\geq SS_0$
and accuracy($\mathcal{A}$) $\geq A_0$
and (significance($\mathcal{A}$) $\geq S_0$
or similarity-score($\mathcal{A}$) $\geq SS_+$)

Figure 2: Compellingness predicate for analogies.

2.2 Name pronunciation

Name pronunciation is taken here to be the task of converting an input spelling (e.g., KEIDEL) into an output pronunciation (**k`ayd´ehl**, which rhymes with MY BELL). The pronunciation is a written specification of how to pronounce the name; it could be fed through a speech synthesizer to produce an actual spoken rendition. A pronunciation includes the phonetic segments or sounds in the name, as well as the level of stress to place on each syllable. Here, the phonetic segments are **kaydehl**, while ` and ´ are stress marks. The ` says to put secondary stress on **kay**. The ´ means primary stress on **dehl**. The notation is taken from DECtalk™[2], but is unimportant for purposes of this paper.

In Anapron, the task of name pronunciation is divided among six principle modules. Table 1 gives a brief account of what each module does, by way of illustration for KEIDEL. The language and morphology modules produce nondeterministic answers; here, the language module generates two possible language classifications of the name — "Generic" or German. This nondeterminism is carried through the other modules until the selection module resolves it by choosing the German analysis. The selection module bases its decision on various rule-based and analogical annotations gathered in the course of analyzing the name under the different language/morphology analyses.

2.3 Application of the method

The method for combining RBR and CBR was applied not to the task of name pronunciation as a whole, but rather to two of its subtasks: transcription and stress assignment. The method requires two main knowledge sources for each subtask: a set of rules, and a case library. The rules for transcription and, to a lesser extent, for stress were based on MITalk (Allen *et al.* 1987) and introductory grammar texts for French, German, Italian, and Spanish. There are 619 transcription rules and 29 stress rules. The case library was derived from a pronouncing dictionary of 5000 names. In addition to

[2]DECtalk is a trademark of Digital Equipment Corporation.

Module	Function	Application to KEIDEL
Language	Determine language	Generic or German
Morphology	Identify prefix, root, and suffix morphemes	KEIDEL = a single root morpheme
Transcription	Map letters to phonetic segments	`kiydehl` if Generic; `kaydehl` if German
Syllable structure	Break into syllables	`kiy-dehl` if Generic; `kay-dehl` if German
Stress assignment	Assign level of stress to each syllable	`k´iydehl` if Generic; `k`ayd´ehl` if German
Selection	Pick best language/ morphology analysis	`k`ayd´ehl` (German)

Table 1: Illustration of Anapron's pronunciation modules for KEIDEL. The output of the modules has been abbreviated for clarity.

these knowledge sources, the method also needs a similarity metric for comparing pairs of cases. The metrics for transcription and stress are based on heuristics about which features of a word determine a given aspect of its pronunciation; for instance, the local spelling context around a letter tends to affect its transcription (due to letter grouping and assimilation effects).

The remainder of this section illustrates how the method was applied to transcription. Consider again the KEIDEL example. In the course of pronouncing this name, Anapron proposes that it is German, and invokes the transcription module under this analysis. The transcription module applies a sequence of operators, each of which converts a string of letters into a string of phonetic segments. It invokes the RC-Hybrid procedure of Figure 1. It starts with K, the first letter of the name.[3] In step (a), the rules suggest the K:`k` operator. This operator maps the letter K to the phonetic segment `k` (as in KITE). No contradictory analogy is found in step (b). Thus in step (c), the operator suggested by the rules, K:`k`, is applied. Application of the next two operators, EI:`ay` and D:`d`, is similarly uneventful, as no contradictory analogies are found.

For the E, things get more interesting. In step (a), the rules suggest E:`ey`, the default pronunciation of E in German (as in FREGE). In step (b), an analogy is found from VOGEL which suggests the E:`eh` operator instead. This analogy has a similarity score of 0.73. Empirical verification reveals that the generalization behind the analogy — which says to apply E:`eh` in German names in a particular context — applies to 7 cases in the case library: EDELBROCK, FOGEL, GEIBEL, LOGEL, SCHNABEL, SPEIDEL, and of course VOGEL. All 7 have E:`eh` applied. Thus the accuracy of the analogy is 7/7 = 1.00. The significance works out to be 0.71. The way the thresholds were set, the analogy is deemed compelling. Thus in step (c), the system selects E:`eh`, overriding the rules by the analogy with VOGEL.

For the final L of the name, the rules suggest L:`l`, which again goes unchallenged. Thus the output of the transcription module for the German analysis of KEIDEL is `kaydehl`.

3 Experiments

The sections below describe the three sets of experiments run on Anapron: the exploratory measurements, the system comparison, and the modification studies.

3.1 Exploratory measurements

Exploratory measurements were taken of Anapron to get a quantitative picture of its operation, and to detect patterns in its behavior that might signal problems. For instance, if the system were found to accept almost all of the analogies that it proposed, this might indicate an overly lax acceptance criterion. In fact, the main result of the exploratory measurements was that the system was being overly strict about accepting analogies. This was shown by an abundance of errors of analogical omission compared to errors of analogical commission.[4]

[3] This exposition is somewhat simplified. In general, the transcription rules are applied in multiple parallel passes, rather than a single, left-to-right pass.

[4] This could be fixed by lowering the system's SS_0 threshold, thereby relaxing the acceptance criterion, or by re-working the similarity metrics to allow better discrimination between good and bad analogies.

The next section describes the test set for this experiment. The subsequent two sections give overviews of the particular measurements made, grouped according to whether they were objective, or included a subjective component.

Test set The test set for this and the other experiments was drawn from the Donnelley corpus, a database of over 1.5 million distinct surnames covering 72 million households in the US. Names in Donnelley range from extremely common (e.g., SMITH, which occurs in over 670,000 households) to extremely rare (e.g., BOURIMAVONG, which occurs in 1 household). The number of households that have a particular name will be referred to as the *frequency* (of occurrence) of the name.

Test sets were constructed from Donnelley by selecting points of interest along the frequency spectrum, and randomly sampling an appropriate number of names at each point.[5] The test set for the objective measurements contained 13 exponentially-distributed frequencies: 1, 2, 4, 8, ..., 4096. The frequencies were distributed exponentially because this yields evenly-spaced measurements of Anapron's behavior — this was determined in a pilot study, which showed that Anapron's percentage of acceptable pronunciations drops linearly as frequency is decreased exponentially. The test set contained a total of 10,000 names, with between 250 and 1000 at each frequency. These numbers represent a tradeoff between the cost of running the test, and the size of the confidence intervals in the resulting measurements. The names were chosen to be disjoint from Anapron's dictionary, since names pronounceable by rote lookup are unrepresentative of system behavior.

Objective measurements Objective measurements were made for both the rule-based and case-based parts of the system. The rule-based measurements counted how many operators were applied by each module (language, morphology, transcription, syllable structure, and stress assignment). The case-based measurements counted how many analogies were proposed, accepted, and rejected, and for what reason (where the reason corresponds to the way the compellingness predicate matched or failed to match the analogy). All measurements were broken down by name frequency, to see how the system's behavior changes as the names get rarer and thus more difficult to pronounce.

The main unexpected finding from the objective measurements was an effect termed the *analogical decline*. It says that as name frequency decreases, the number of highly plausible analogies[6] to the name also decreases; however, the overall number of analogies (highly plausible or otherwise) does not decrease significantly. This asymmetric decrease in analogical activity is investigated further in Golding (1991).

Subjective measurements Subjective measurements of the system's behavior were made not on the 10,000-name test set described above, but on a scaled-down 1,000-name version. This was necessary to make it feasible to obtain human judgements. The 1,000-name test set had 250 names at each of four (roughly) exponentially-distributed frequencies: 1, 32, 256, and 2048.

The subjective measurements consisted of judgements, for each name, about the acceptability of the following: the overall pronunciation, the individual transcription and stress operators applied, the choice of language/morphology analysis, and the analogies proposed (whether accepted or rejected). The judgements were made by the first author. To facilitate this rather laborious process, a *judgement editor* was used, which provided a graphical user interface for entering or changing judgements about a name. The editor also verified that the judgements for a name were complete and consistent.

The main result of the subjective measurements was that errors of analogical omission were found to be far more numerous than errors of analogical commission. This suggests that the system's analogical acceptance criterion may have been too conservative.

3.2 System comparison

To see how the combined RBR/CBR approach performs relative to other methods, Anapron was compared with seven other name-pronunciation systems: three state-of-the-art commercial systems (from Bellcore, Bell Labs, and DEC), two versions of a machine-learning system (NETtalk[7]), and two humans. The sections below describe the test set, design, and analysis of the experiment. For a fuller presentation, see Golding and Rosenbloom (1993).

Test set The test set for the system comparison was similar to that used in the subjective measurements, except that: (1) only 100 names (not 250)

[5] If Donnelley had fewer than the desired number of names at some frequency f, then the names were selected randomly from the narrowest symmetric frequency band around f that was big enough.

[6] A highly plausible analogy is one whose similarity score is SS_+ or greater.

[7] The two versions of NETtalk will be referred to as BP-legal and BP-block. BP-legal is vanilla NETtalk; BP-block is NETtalk enhanced with a "block decoding" postprocessor (Dietterich *et al.* 1990).

System	Name frequency				Overall
	2048	256	32	1	
Ubound	98	98	98	96	97
Human1	97	93	93	88	93
Human2	98	94	94	86	93
Comm1	97	95	93	90	93
Comm2	96	90	87	86	90
Comm3	96	94	89	78	89
Anapron	91	88	85	80	86
BP-block	84	83	77	69	78
BP-legal	78	72	66	52	67

Table 2: Percentage of acceptable scores for each system, broken down by name frequency.

were chosen at each frequency, to reduce the burden on the human test subjects; and (2) the test set was no longer constrained to be disjoint from Anapron's dictionary, as an unbiased measurement of system performance includes names both in and out of the dictionary.

Design The first step of the experiment was to run each system on the 400-name test set. The output of the computer systems was gathered in the form of written pronunciations (before they were sent to a speech synthesizer). The output of the humans was tape-recorded and transcribed as written pronunciations. In the case of NETtalk, the system also needed to be trained; this was done using Anapron's 5000-name pronouncing dictionary.

A cassette tape was then made of the pronunciations. To hide the identities of the systems, all pronunciations were read by the DECtalk speech synthesizer. For each name, duplicate pronunciations were eliminated, and the remaining pronunciations were permuted randomly. The order of names was permuted randomly as well. A set of 14 human subjects listened to the cassette tape and rated the acceptability of each pronunciation.

Analysis The main results of the system comparison appear in Table 2.[8] It gives the percentage of acceptable scores for each system, broken down by name frequency. The table includes an imaginary ninth system, labelled Ubound, which generates for each name the pronunciation that received the greatest number of acceptable votes from the judges. It measures the degree to which all judges can be pleased simultaneously, using just the pronunciations available from the eight systems tested.

[8]The names of the commercial systems and humans have been omitted since this paper is concerned with evaluation methodology rather than the results per se.

System	Name frequency				Overall
	2048	256	32	1	
Human1	+	+	+	+	+
Human2	+	+	+	+	+
Comm1	+	+	+	+	+
Comm2	+	+?	+?	+	+
Comm3	+	+	+	−?	+
BP-block	−	−	−	−	−
BP-legal	−	−	−	−	−

Table 3: Differences in performance between other systems and Anapron, broken down by name frequency. A plus sign (+) means higher acceptability than Anapron; a minus sign (−) means lower acceptability. All differences are significant at the 0.01 level, except those marked with a question mark (?), which are not significant even at the 0.10 level.

Table 2 shows that Anapron performs almost at the level of the commercial systems, and substantially better than the two versions of NETtalk. Also, although the eight systems seem to hit a performance asymptote at 93%, the Ubound system demonstrates that it is possible to score at least 97%. This suggests that there is room for improvement in all systems.

To detect whether the differences between Anapron and the other systems were statistically significant, an ANOVA was run, followed up by a Bonferroni multiple comparison procedure. Table 3 gives the results. It shows that overall, Anapron outperformed the two versions of NETtalk, but the commercial systems and humans did better than Anapron. However, in some frequency ranges, a significant difference between Anapron and certain commercial systems could not be detected.

3.3 Modification studies

To gauge the contribution of Anapron's components to its overall performance, a set of experiments were performed that modified the components and observed the effects on system performance. There were five such studies, modifying: rules and cases, thresholds, language knowledge, morphology knowledge, and syllable-structure knowledge. The first study — on rules and cases — addressed the central hypothesis of the system concerning the efficacy of combining rules and cases. It showed that the system achieved higher accuracy by combining the two than it could have achieved with either one alone. The threshold study tested how sensitive the system's performance was to the threshold settings used in the definition of analogical compellingness — i.e., SS_0,

SS_+, A_0, and S_0 (see Figure 2). Extreme raising or lowering of any one threshold at a time was generally found to hurt accuracy, although lowering of SS_0 sometimes improved accuracy at the expense of increasing run time. The remaining three studies concerned the system's support knowledge — i.e., knowledge needed in service of the two top-level tasks, transcription and stress. Degrading the language or morphology knowledge sufficiently was found to have a substantial negative impact on system accuracy, while degrading syllable-structure knowledge had a relatively minor effect.

The sections below focus on the first of these experiments — the rule/case study. They discuss the test set, design, and analysis of the study.

Test set Like the system comparison, the rule/case experiment required a great deal of human effort in the evaluation. The test set was therefore made the same size as in the system comparison — 100 names at each of four frequencies. The only difference was that, as in the exploratory measurements, the test set was constrained to be disjoint from Anapron's dictionary, since again rote lookup behaviors were not of interest.

Design The rule/case study involved independently varying the strength of the system's rules and cases. For each combination of rule strength and case strength, the system was run on the 400-name test set, and its accuracy and run time were recorded. Accuracy was measured as the proportion of acceptable pronunciations generated by the system, where acceptability was judged by the first author.[9] All judgements were cached and re-used if a pronunciation recurred, to help enforce consistency across trials.

The rules were set to four different strengths: 0, 1/3, 2/3, and 1. A strength of 1 means all transcription and stress rules were retained in the system. Strength 0 means that all rules were deleted except *default* rules. The default rules transcribe a letter or assign stress if no other more specific rule matches. The default rules cannot be deleted, otherwise the system would be unable to generate a complete pronunciation for some names. Retaining the default rules corresponds to keeping 137 out of 619 transcription rules and 16 out of 29 stress rules. As for rule strengths between 0 and 1, these correspond to retaining a proportional number of non-default rules in the system. Each strength is obtained by deleting a random subset of the non-default rules from the next higher strength.

[9] The first author was an unusually harsh judge, thus the scores here are not directly comparable to those of the system comparison.

Rule strength	Case strength 0	1000	2000	3000	4000	5000
0	19	27	32	34	35	36
1/3	33	39	43	44	46	47
2/3	46	54	56	56	57	59
1	56	65	65	67	67	68

Table 4: System accuracy results. Each value is the percentage of names in the test set for which the system produced an acceptable pronunciation.

The cases were set to six strengths: 0, 1000, 2000, 3000, 4000, and 5000. The strength is just the number of names that were kept in the case library. Again, each weakening of the case library produces an arbitrary subset of the previous case library.

Analysis Although accuracy and run-time data were both collected, only the accuracy results will be reported here. For the run-time results, see Golding and Rosenbloom (1991).

Table 4 shows system accuracy as a function of both rule strength and case strength. The main result is that accuracy improves monotonically as rule or case strength increases. The total improvement in accuracy due to adding rules is between 32% and 38% of the test set (depending on case strength). For cases it is between 12% and 17% (depending on rule strength). This substantiates the central hypothesis about the system — that by combining rules and cases, it can achieve a higher accuracy than it could with either one alone.

4 Lessons Learned

Two kinds of lessons emerge from the evaluation presented here: lessons for CBR evaluation methodology, and lessons for CBR theory. These are discussed below, together with issues that arose in designing the evaluation.

Lessons for CBR evaluation methodology The work presented here suggests a three-part evaluation methodology: (i) Profile the operation of the system to check for unexpected behaviors; (ii) Do a system comparison to gauge the overall performance of the system — this provides at least indirect evidence that the methods under investigation are sound, in that they lead to high performance; and (iii) Run modification studies to understand the contribution made by each part of the system.

Expressed at this abstract level, this evaluation methodology applies not only to CBR, but to a wide range of computer systems. The instantiation to Anapron showed more about how to evaluate

CBR systems in particular. For instance, part (i) of Anapron's evaluation counted errors of analogical omission and commission, a measure that is relevant for most case-based systems. In part (iii) of Anapron's evaluation, the first experiment broke down the system into fairly coarse components: RBR and CBR. This was because the main hypothesis concerned these components. The same methodology could be applied just as well to lower-level components, such as retrieval or adaptation, if these are the components of interest.

Lessons for CBR theory The first lesson learned from the Anapron experiments for CBR theory is that the accuracy of CBR systems can be improved by combination with RBR. This comes directly from the confirmation of the main hypothesis about the system.

A second lesson is that the compellingness predicate (and associated empirical verification) gives the system the ability to weed out bad analogies. This is an important ability for any CBR system that has the potential to draw incorrect analogies. The effectiveness of the compellingness predicate was demonstrated in part (iii) of the experiments, which showed that substantial tampering with even a single compellingness threshold at a time generally hurt system accuracy.

A third lesson is that the introduction of RBR provides a convenient and natural way to index cases: prediction-based indexing. While PBI was not directly tested in the experiments, indirect evidence for its success comes from part (ii), which showed that the system, using this indexing method, achieves very high performance. An additional benefit of PBI, not measured in the experiments, is in saving development time — it frees the system designer from having to analyze the domain, identify a good indexing vocabulary, devise appropriate computed features, and so on. Instead, PBI exploits the structure of the domain that is implicit in the rules and thus already available.

A fourth lesson for CBR theory is that rational reconstruction affords a novel method of case adaptation. Traditional adaptation techniques do some kind of repair to make the source and target cases compatible. The approach in Anapron is to use RR to break down a source case into the individual operators that were applied to solve it. These individual operators are then fine-grained enough to be transferred verbatim from source to target. This can be thought of as "adaptation by factoring into operators". As with PBI, the experiments reported here do not directly show the benefits of this adaptation strategy, but the system comparison shows that it leads to high overall performance.

Design issues In designing the Anapron experiments, a few issues arose both for evaluating systems in general, and for evaluating CBR systems in particular. The main general issue was obtaining the various forms of knowledge needed to run the experiments: a test set; human judgements of each system's answers to the test set; and, in certain cases, auxiliary knowledge bases, such as an aligned dictionary of names and their pronunciations for training NETtalk. System evaluations would be greatly facilitated if this sort of knowledge were already commonly available. Shared datasets, such as the U.C. Irvine Repository of Machine Learning, are a good start in this direction, but more effort is still needed.

Two CBR-specific issues arose in Anapron's evaluation, and remain open problems. First, it would be desirable to run a modification experiment on the similarity metric; however, it is not obvious how to characterize the space of possible similarity metrics. Second, while it is relatively straightforward to count errors of analogical commission (harmful analogies), it is harder to detect all errors of analogical omission (helpful analogies that were missed).

5 Acknowledgements

We would like to thank Mark Liberman, Cecil Coker, Murray Spiegel, Tony Vitale, Tom Dietterich, John Laird, Connie Burton, and David Shapiro for their help with the system comparison. We are also indebted to Murray Spiegel for arranging for us to borrow a pronouncing dictionary of names from Bellcore.

References

J. Allen, M. S. Hunnicutt, and D. Klatt. *From Text to Speech: The MITalk System*. Cambridge University Press, Cambridge, 1987.

T. G. Dietterich, H. Hild, and G. Bakiri. A comparative study of ID3 and backpropagation for English text-to-speech mapping. In *Proceedings of 7th IMLW*, Austin, 1990. Morgan Kaufmann.

A. R. Golding. *Pronouncing Names by a Combination of Rule-Based and Case-Based Reasoning*. PhD thesis, Stanford University, 1991.

A. R. Golding and P. S. Rosenbloom. Improving rule-based systems through case-based reasoning. In *Proceedings of AAAI-91*, Anaheim, 1991.

A. R. Golding and P. S. Rosenbloom. A comparison of Anapron with seven other name-pronunciation systems. *Journal of the American Voice Input/Output Society*, 14, 1993.