Memory-Based Reasoning Applied to English Pronunciation

Craig W. Stanfill
Thinking Machines Corporation
245 First Street
Cambridge, MA 02142

Abstract

Memory-based Reasoning is a paradigm for AI in which best-match recall from memory is the primary inference mechanism. In its simplest form, it is a method of solving the inductive inference (learning) problem. The primary topics of this paper are a simple memory-based reasoning algorithm, the problem of pronouncing english words, and MBRtalk, a program which uses memory-based reasoning to solve the pronunciation problem. Experimental results demonstrate the properties of the algorithm as training-set size is varied, as distracting information is added, and as noise is added to the data.

I. Introduction

The *Memory-Based Reasoning Paradigm* [Stanfill and Waltz, 1986] places recall from memory at the foundation of intelligence. Simply stated, the paradigm assumes that much of what can be explained as common sense or learned behavior can be explained in terms of recall from memory. For example, faced with a rainy day, a human might remember similar days in the past, and note that on days when he had an umbrella he did not usually get wet, while on days when he had no umbrella he got very wet indeed.

The principle operation of memory-based reasoning is retrieving "the most relevant item" from memory[1]. This requires an exhaustive search which, on a sequential machines, is prohibitively expensive for large databases. The only alternative is to index the database in a clever way (e.g. [Kolodner, 1980]). No truly general indexing scheme has yet been devised, so the intensive use of memory in reasoning has not been extensively studied. The recent development of the Connection Machine[2] System [Hillis, 1985] has changed this situation: a CMS is capable of applying an arbitrary measure of relevance to a large database and retrieving the most relevant items in a few milliseconds.

The first use of memory-based reasoning has been for the *inductive inference* task. Given a collection of data which has been partitioned into a set of disjoint classes (the *training data*) and a second collection of data which has not been classified (the *test data*), the task is to classify the test data according to patterns observed in the training data. To date, this task has been worked on in the connectionist paradigm (e.g. backpropagation learning [Sejnowski and Rosenberg, 1986],) the rule-based paradigm (e.g. building decision trees [Quinlan, 1979]), and the classifier-system paradigm (e.g. genetic algorithms [Holland *et al.*, 1986]).

Experiments conducted over the last year now solidly confirm the applicability of memory-based reasoning to inductive inference. A program called MBRtalk, operating within the memory-based reasoning paradigm, has demonstrated strong performance on the task of inferring the pronunciation of english words from a relatively small sample. MBRtalk infers the pronunciation of novel words, given only a dictionary of 18,098 words. On a phoneme-by-phoneme basis, it is correct approximately 88% of the time. Furthermore, performance degrades gracefully, so that the pronunciation it generates is almost always plausible.

II. Background

The most intensively studied setting of the inductive inference mode problem occurs in the rule-based systems paradigm, where it goes under the name "similarity-based learning[3]." Here it takes the form of learning a set of rules from a collection of training data. For a recent survey, see [Carbonell *et al.*, 1983]. For more in-depth treatments, see [Michalski *et al.*, 1983] and [Michalski *et al.*, 1986].

There is a closely related line of research which goes under the name *case-based reasoning* (see, e.g. [Kolodner, 1985] [Lehnert, 1987]). It is similar to memory-based reasoning in that recall from memory plays a role in learning, but different in that it presupposes substantial knowledge

[1] A computational measure of relevance is the essence of implementing MBR.

[2] Connection Machine is a registered trademark of Thinking Machines Corporation.

[3] There is also "model-based" learning, which depends on the learner having a substantial amount of knowledge about the problem at hand.

about the target domain in the form of a deductive procedure. In addition, case-based reasoning operates within the rule-based paradigm, so that whatever knowledge is extracted from cases is stored in a rule-like form.

Inductive inference has also been studied in the connectionist paradigm. Specifically, in backpropagation learning [Sejnowski and Rosenberg, 1986], classification is accomplished by a three-layer network, with weighted links connecting adjacent layers. Learning is accomplished by running the network against the training data, generating an error signal, and adjusting link weights.

A *Classifier System* [Holland *et al.*, 1986] is a collection of primitive classification rules, each consisting of a condition-action pair. Initially, the system contains random classifiers. Learning takes place through an evolutionary process, typically including genetic crossover and mutation operations; the right to reproduce is governed by the success of the classifier in correctly classifying the data.

III. Pronunciation as a Test Domain

Memory-based reasoning has been tested on the pronunciation problem: given the spelling of a word, determine its pronunciation. The training data for this problem is a dictionary, and the test data is a set of words not in that dictionary. There are a number of advantages to working in this domain. First, training data is available in large quantities. Second, the domain is rich and complex, so that any inductive algorithm is sure to be tested rigorously.

Unfortunately, perfect performance is fundamentally impossible. First, there are irregularities in english pronunciation, so that some words must always be learned by rote. Second, some words have different pronunciations, depending on whether they are used as nouns or verbs:[4]

live = "līv" or "liv"
object = "ôb′jϵkt" or "ϵb jekt′"

Third, many words have several allowable pronunciations regardless of how they are used:

amenity = "ϵ men′ ϵ tē" or "ϵ mē′ nϵ tē"

Fourth, many words of foreign origin have retained foreign pronunciations:

pizza = "pēt′sϵ" vs fizzy = "fi′zē"
montage = môn täzh′ vs. frontage = "frunt′ϵj"

The pronunciation task has been studied within the connectionist paradigm [Sejnowski and Rosenberg, 1986]. Backpropagation learning was applied to a transcription of speech by a child and to a subset (1000 words) of Webster's dictionary [Webster, 1974]. In each case, both a text and a phonetic transcription of that text was repeatedly presented to a network. The experiment was primarily evaluated according to how well it could reproduce the phonetic transcription given only the text — no novel text was introduced. Thus, although this experiment provides important insight into the properties of backpropagation learning as a form of self organizing system, the results are not directly comparable to those from MBRtalk.

The pronunciation task has also been studied in the case-based reasoning paradigm [Lehnert, 1987], with results similar to those reported below.

IV. Representation

In order to apply memory-based reasoning to pronunciation, it is necessary to devise a representation for the words in the dictionary. The representation used in MBRtalk is identical to that used in NETtalk[5]. For every letter of every word in the database, we create a "frame," which consists of the letter, the previous four letters, the succeeding four letters, the phoneme corresponding to that letter, and the stress assigned to that letter.

Certain difficulties are associated with this representation, primarily due to the fact that the correspondence between letters and phonemes is not one-to-one. First, two letters sometimes produce a single phoneme, as the double 's' in 'kiss' (kis-). This is handled by using the letter '-' as a silent place holder. Second, the existence of diphthongs and glides may cause one letter to yield several phonemes, as the first 'u' in 'future' (fyo͞ochᵋr). This problem is solved by treating dipthongs as if they were single phonemes, so that the 'u' in 'future' becomes 'yo͞o'. Finally, stress is not indicated by an accent mark, but by a separate stress field, which can contain the '0' for unstressed vowels, '1' for primary stress, '2' for secondary stress, and '+' or '-' for consonants (rising and falling stress). Applying these principles, we get the following transcription for 'future':

Text	f	u	t	u	r	e
Phonemes	f	yo͞o	ch	-	ᵋr	-
Stress	+	1	-	-	0	-

As noted above, each letter of each word yields a *frame* consisting of the letter, the four preceding letters, the four succeeding letters, plus the phoneme code and the stress code corresponding to the letter. These fields are called *n-4* through *n-1* (the preceeding four letters); *n* (the letter itself); *n+1* through *n+4* (the suceeding four letters); *p*

[4] The phonological symbols used below correspond to common usage in dictionaries. Due to font limitations, the symbol 'ϵ' is used to stand for the unstressed vowel sound usually represented by a schwa.

[5] With the exception that NETtalk used a 7-letter window, while we use a 9-letter window.

(the phoneme); and *s* (the stress). Thus, the word 'future' yields the following 6 frames:

	f	utur	f	+
f	u	ture	y$\overline{oo}$	1
fu	t	ure	ch	-
fut	u	re	-	-
futu	r	e	$^{\epsilon}$r	0
utur	e		-	-

V. Metrics for Memory-Based Reasoning

To implement memory-based reasoning, we need a computational measure of similarity. This section will first explain some notation, then present two different metrics which have been used in pronunciation experiments.

A *record* is a structured object containing a fixed set of *fields*. A field may be empty, or it may contain a value. A *database* is a collection of records[6].

A *target* is a record containing some empty fields. The empty fields are called *goals*, and the non-empty fields are called *predictors*.

A *metric* is a function giving the dissimilarity of two records. The number of possible metrics is immense, and no claim is being made that the following two metrics are optimal.

We compute the dissimilarity between two records by assigning a penalty for each field in which they differ. For example, if we have the following two frames:

f	u	ture	y$\overline{oo}$	1
n	u	ture	$\overline{oo}$	1

we would compute their dissimilarity by assessing a penalty for the field *n-1*.

The penalty function we are using is based on how tightly a single field-value pair in a predictor field constrains the value of the goal field. For example, the field-value pair [*n*='b'] gets a high weight because, if field *n* contains a 'b', the the phoneme field can only contain 'b' or '-'. On the other hand, the field-value pair [*n-4*='a'] gets a low weight because, if field *n-4* (the fourth previous letter) contains a 'a', the phoneme field might contain almost anything. The exact form of this penalty function is contained in [Stanfill and Waltz, 1986].

There are two variations on this metric: we can use the penalty function based on the contents of the target record or of the data record[7]. In the example above, we might use the penalty function associated with [$n - 1$ = 'f'] or with [$n - 1$ = 'n']. If the penalty function depends on the target record we have a *uniform metric*. If the penalty function depends on the data record we have a *variable metric*.

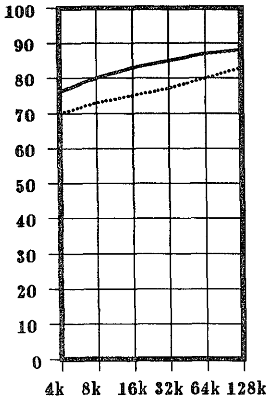

Figure 1: Database Size

VI. Experimental Results

These two metrics were applied to the pronunciation task, and their sensitivity to database size, distraction, and noise was determined.

The first task was to determine how the quality of the pronunciation varied as size of the database changed. The raw databases consisted of frames generated from Webster's dictionary [Webster, 1974]. First, 1024 frames were extracted and set aside as test data. Second, various quantities of training data were extracted; the smallest sample was 4096 frames and the largest was 131,072. Memory-based reasoning, using the two different metrics noted above, was applied to the test data. The value MBR predicted for the phoneme slot was then compared with the value already stored there. With the largest database, using the uniform metric, the accuracy rate was 88%. Using the variable metric, the best accuracy was 83%. The performance of both algorithms degraded gracefully as the size of the database was reduced. With a sample of only 4K frames (approximately 700 words), MBR still managed to get the correct answer 76% of the time (Figure 1).

The next task was to determine how well the two algorithms rejected spurious information (distraction). This was done by adding between 1 and 7 fields containing

[6] Duplicates may be present.

[7] A third alternative is to use a penalty function depending on both values.

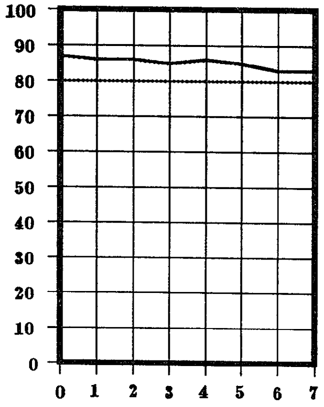

Figure 2: Distraction

random values to each frame in a 64K-frame training database. The uniform metric's performance degraded slightly, from 88% down to 83% correct. The variable metric's performance did not change at all (Figure 2).

The third task was to determine how well the metrics performed in the presence of noise. Two different types of noise must be considered: predictor noise and goal noise. N-% noise is added to a field by randomly choosing N-% of the records, then giving them a randomly selected value.[8] In the predictor-noise test, a fixed percentage of noise was added to every predictor field in a 64K-record training database, and the results tabulated. The uniform metric was relatively unaffected: with 90% noise, performance declined from 88% to 79%, after which it quickly dropped to chance. The variable metric was somewhat surprising: performance was actually *better* with 10% - 50% noise than with none (Figure 3).[9]

When noise was added to goal fields, both algorithms' performances dropped off more-or-less linearly (Figure 4).

In summary, for the pronunciation task the uniform metric is always more accurate than the variable metric. It has fairly good resistance to distraction, and extremely good resistance to predictor noise. It does not resist goal noise particularly well. The variable metric does, however have some useful properties: it seems immune to distrac-

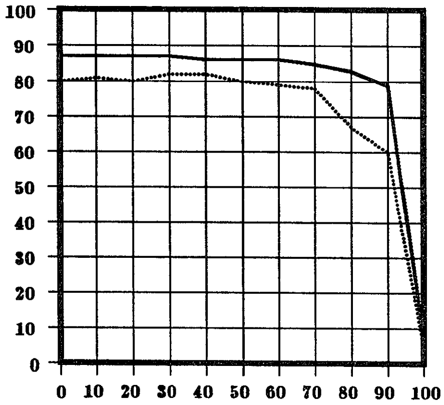

Figure 3: Predictor Noise

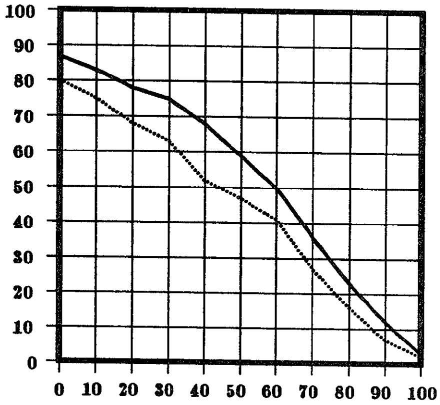

Figure 4: Goal Noise

[8] These values were uniformly distributed. An alternative experiment would have been to select a random value having the same distribution as the data occuring in the field.

[9] For a discussion of the effect of noise on concept learning, see [Quinlan, 1986].

tion, and has even better resistance to predictor noise. The anomalous improvement in performance as predictor noise increases up to 50% needs to be understood.

VII. Discussion

Substantial work remains to be done on the mechanics of memory-based reasoning. First, a variety of metrics need to be studied. Second, MBR needs to be extended to work in domains with continuous variables. Third, tasks other than pronunciation need to be attacked. Fourth, there is a need for research into the effects of representation on learnability. Finally, a rigorous head-to-head comparison between MBR and other methods of inductive inference is needed.

The most striking aspect of this experiment is high performance on a difficult problem with a very simple mechanism. There are no rules, and neither complex algorithms nor complex data structures. If simplicity is a good indicator of the plausibility of a paradigm, then memory-based reasoning has a lot going for it.

The ultimate goal of Memory-based Reasoning remains to build intelligent systems based on memory. This experiment is an important first step in that direction. What has been demonstrated is that it is possible to use memory as an inference engine; that if an agency can store up experiences and then recall them on a best-match basis, it can learn to perform a complex action. Much remains to be done, but the memory-based reasoning paradigm has passed a crucial first test.

Acknowledgements

Many thanks to Dave Waltz, who is the co-originator of the Memory-based reasoning paradigm; to George Robertson who got learning research going at Thinking Machines Corporation; to Donna Fritzsche and Robert Thau for assisting with software; to Danny Hillis for designing a very nice machine; and to Thinking Machines Corporation for supporting this research.

References

[Carbonell *et al.*, 1983] Jaime Carbonell, Ryszard Michalski, and Tom Mitchell. Machine Learning: A Historical and Methodological Analysis. *AI Magazine* 4(3):69-79, 1983.

[Hillis, 1985] Danny Hillis. *The Connection Machine.* MIT Press, Cambridge Massachusetts, 1985.

[Holland *et al.*, 1986] John Holland, Keith Holyoak, Richard Nisbett, and Paul Thagard. *Induction: Processes of Inference, Learning, and Discovery.* MIT Press, Cambridge Massachusetts, 1986.

[Kolodner, 1980] Janet Kolodner. "Retrieval and Organizational Strategies in Conceptual Memory: A Computer Model." Technical Report 187, Yale University, Department of Computer Science, 1980 (Ph.D. Dissertation).

[Kolodner, 1985] Janet Kolodner and Robert Simpson. "A Process Model of Case-Based Reasoning in Problem Solving." In *Proceedings IJCAI-85*, Los Angeles, California, International Joint Committee for Artificial Intelligence, August 1985.

[Lehnert, 1987] Wendy Lehnert. Case-Based Problem Solving with a Large Knowledge Base of Learned Cases. In *Proceedings AAAI-87*, Seatle, Washington, American Association for Artificial Intelligence, 1987.

[Michalski *et al.*, 1983] Ryszard Michalski, Jaime Carbonell, and Tom Mitchell, editors. *Machine Learning.* Morgan Kaufman, Los Altos, California, 1983.

[Michalski *et al.*, 1986] Ryszard Michalski, Jaime Carbonell, and Tom Mitchell, editors. *Machine Learning*, Volume 2. Morgan Kaufman, Los Altos, Californai, 1986.

[Quinlan, 1979] Ross Quinlan. "Discovering Rules from Large Collections of Examples: A Case Study." In *Expert Systems in the Micro Electronic Age.* Donald Michie, editor. Edinburgh University Press, Edinburgh, 1979.

[Quinlan, 1986] Ross Quinlan. "The Effect of Noise on Concept Learning." in *Machine Learning,* Volume 2. Ryszard Michalski *et. al.,* editors. Morgan Kaufman, Los Altos, Californai, 1986.

[Sejnowski and Rosenberg, 1986] Terry Sejnowski and Charley Rosenberg. "NETtalk: A Parallel Network that Learns to Read Aloud." Technical Report JHU/EECS-86, The Johns Hopkins University Electrical Engineering and Computer Science Department.

[Stanfill and Waltz, 1986] Craig Stanfill, and David Waltz. "Toward Memory-Based Reasoning." *Communications of the ACM* 29(12):1213-1228, December 1986.

[Webster, 1974] *Merriam Webster's Pocket Dictionary,* 1974.