Hidden Markov Models for Coding Story Recall Data

Michael A. Durbin (golden@utdallas.edu)
Cognitive Science Program (Attention: Professor Golden)
University of Texas at Dallas, GR4.1, Box 830688
Richardson, TX 75083-0688

Jason Earwood (golden@utdallas.edu)
Psychology Program (Attention: Professor Golden)
University of Texas at Dallas, GR4.1, Box 830688
Richardson, TX 75083-0688

Richard M. Golden (golden@utdallas.edu)[1]
Psychology and Cognitive Science Programs, GR4.1, Box 830688
University of Texas at Dallas
Richardson, TX 75083-0688

Abstract

Current methods of coding recall, summarization, talk-aloud, and question-answering data are inherently unreliable and not effectively documented. If the process of coding protocol data could even be partially automated, this would be an important scientific advance in the field of text comprehension. Twenty-four human subjects read and recalled each of four short texts. Half of the human recall data (the "training data") was coded by a human coder and then used to estimate the parameters of a set of Hidden Markov Models (HMMs) where each HMM was associated with a particular complex proposition in the text. The Viterbi algorithm was then used to assign the "most probable" complex proposition to human-coder specified text segments in the remaining half of the human recall data (the "test data"). The HMM algorithm made coding decisions which agreed well with a human coder's decision on the test data indicating that the HMM is indeed capable of formally representing a human coder's "theory" of how text segments should be mapped into complex propositions for simple texts.

Introduction

Theories and experiments in the field of text comprehension often require mapping recall (e.g., Golden, 1997), summarization (e.g., van den Broek & Trabasso, 1986), talk-aloud (e.g., Trabasso & Magliano, 1996), and question-answering (e.g., Graesser & Franklin, 1990) protocol data into a semantic model of the implicit and explicit information in text clauses. This semantic model of the information in the text clauses has been referred to by Kintsch (1998) as the *textbase microstructure*. Typically this initial *coding procedure* of mapping the protocol data into a textbase microstructure is done using human coders. Inter-coder reliability measures are then used to establish the reliability of the coding procedure.

This widely used coding procedure methodology, however, has several problems. First, such coding procedures are typically not well documented. Second, the reliability of such procedures is often highly dependent upon "human coders", who despite their best intentions, are prone to inconsistent coding behaviors (especially over very large coding tasks). Third, such coding procedures are typically not readily accessible to other researchers. And fourth, coding procedures across research labs located in different parts of the world are not standardized in any particular manner.

An ideal solution to these problems would be to develop an automated approach to coding human protocol data (as advocated by Ericsson and Simon, 1993). Although important progress in this area has been made (see especially Kintsch, 1998, Chapter 3), additional work is required. It should also be emphasized that the task of coding human protocol data is not nearly as complex as the full-fledged natural language understanding problem. Consider a typical experiment where a group of human subjects are asked to recall the same story from memory. Although the resulting protocol data will be extremely rich and varied, typically the text comprehension researcher is only interested in detecting a relatively small number of complex propositions. This dramatically simplifies the pattern recognition problem.

The main goal of this research is to develop and empirically evaluate a new theoretical framework for reliably mapping protocol data into a textbase microstructure. Specifically, a Hidden Markov Model (HMM) (see Allen, 1995; Charniak, 1993; Jelinek, 1997; for relevant reviews) is constructed for each complex proposition in each of four short stories. The stories, based upon classic fables, each consisted of approximately 10-15 short sentences with each sentence corresponding roughly to a complex proposition (Golden, 1997). Twenty-four human subjects read and recalled each of the four short texts (see Golden, 1997, for additional details). Half of the human recall data (the "training data") was coded by a human coder and then used to estimate the parameters of the HMM associated with each

[1] The order of the authors is arbitrary. Please address all correspondence to Richard M. Golden.

complex proposition. The prior probability that a particular complex proposition was used by the human coder was also recorded. Next, the Viterbi algorithm (Viterbi, 1967; see Allen, 1995; Charniak, 1993; Jelinek, 1997) was used to assign the "most probable" complex proposition to human-coder specified text segments in the remaining half of the human recall data (the "test data"). Measures of agreement between the human coder and AUTOCODER were then computed using only the test data. A high measure of agreement indicates that the HMM is indeed capable of formally representing a human coder's "theory" of how text segments should be mapped into complex propositions.

Method

Human Protocol Data

Texts. The human protocol data used consisted of recall data associated with four texts collected by Golden (1997). The four texts ("Cuckoo", "Miser", "Eagle", and "Doctor") were especially written to have approximately similar levels of syntactic and semantic complexity. Each sentence in the text was written to conform approximately to: (1) a standard subject-verb-object form, and (2) such that each sentence corresponded roughly to one complex proposition. For example, the "Miser" text read by the human subjects is shown below.

The "Miser" Text (Golden, 1997)

```
A miser bought a lump of gold using
all of his money. The miser buried the
gold in the ground. The miser looked at
the buried gold each day. One of the mi-
ser's servants discovered the buried
gold . The servant stole the gold . The
miser , on his next visit , found the
hole empty . The miser was very upset .
The miser pulled his hair . A neighbor
told the miser not to be upset . The
neighbor said , " Go and take a stone ,
and bury it in the hole . "The neighbor
said , " And imagine that the gold is
still lying there ." The neighbor said ,
" The stone will be as useful to you as
the gold . " The neighbor said, " When
you had the gold , you never used it . "
```

Recall Protocol Data. Twenty-four college students read and verbally recalled each of four texts ("Miser", "Cuckoo", "Doctor", and "Eagle") from memory as described in Golden (1997). The recall data was then transcribed. Text segments in all of the recall protocol data corresponding to complex propositions were then identified by human coders. The recall data from twelve of the college students was designated as *training data*, while the recall data from the remaining twelve college students was designated as *test data*.

To provide some insights into the richness and complexity of the statistical pattern recognition problem considered in this paper. Here is an example recall protocol extracted from the training data set.

Subject 1 recall of "Miser Text"
(training data set)

```
someone that a servant that knew that
discovered the money# and took it# and
then the miser saw that the money was
gone# and he was upset# and complained
to a neighbor# and the neighbor said
well just get a stone and bury your
money# dig a hole and bury the money#
because it'll do you just as much good
as your real money your gold is doing
you#
```

The symbol # in the above recall protocol associated with subject 1 refers to the marking of text segments by an experienced human coder. Text segments corresponding to complex propositions were marked by experienced human coders for both the training data and test data sets. Here is a representative recall protocol from subject 12 who was assigned to the test data set. The complexity of the recall data (even when a human coder has already identified text segments) is readily apparent (compare recall data of Subject 1, Subject 12 with one another and the original "Miser" text).

Subject 12 recall of "Miser Text"
(test data set)

```
and he buried it in the ground # and he
went over every day to look at where the
money was where the lump of gold was
buried# and one day when the miser was-
n't there a thief came and dug up the
lump of gold# and so the miser goes and
he sees the hole in the ground# and he's
very upset by that# and a bystander
tells the miser to take a rock and bury
it in the ground# and the miser says
why# and the bystander says well all you
ever did was look at the ground anyway#
you never did use the gold# so there
might as well be a rock there#
```

Parameter Estimation (Learning Algorithm)

The learning process involves a specially designed graphical user-interface which is referred to as AUTOCODER. Figure 1 shows a typical AUTOCODER display. A subject's recall data (in this case, the recall data for Subject 12) is displayed. The human coder first segments the text so that each word sequence in each text segment corresponds to a complex proposition. Beneath each word is a pull-down menu consisting of a series of concepts. The human coder decides which words (or word sequences) should be assigned concepts, and then uses the pull-down menu to assign a concept to each selected word within a given text segment. Another pull-down menu is then used to

assign a complex proposition to a given sequence of concepts within a text segment.

Probabilistic Modeling Assumptions. Let W_1, ..., W_M be the ordered sequence of words (or more generally word phrases) within a particular text segment which an experienced human coder has decided should be assigned concepts. Let C_i denote the concept assigned to the ith word, W_i. Let F be the complex proposition assigned to the concept sequence C_1, ..., C_M.

After the human coder has completed the coding task, AUTOCODER has stored the following items for the human coder. First, a concept dictionary consisting of the concepts created by the human coder. Second, a complex proposition dictionary consisting of the complex propositions created by the human coder. Third, the percentage of times that a particular complex proposition F has been used (denoted by $p(F)$). Fourth, the percentage of times that a word (or word phrase) W_i is used to express the concept C_i (denoted by $p(W_i \mid C_i)$) is computed (this is referred to as the "emission probability" in the HMM literature). And fifth, the percentage of times that one concept follows another concept given a particular complex proposition F (denoted by $p(C_{i+1} \mid C_i, F)$) (this is referred to as the "transition probability" in the HMM literature). Given the usual conditional independence assumptions of an HMM, these statistics in conjunction with the concept and complex proposition dictionaries correspond to a particular type of probabilistic theory of how the human coder codes the recall data.

For example, consider the text segment *"He buried his life savings deeply in the ground"*. The human coder might choose to model this text segment as an ordered sequence of word phrases: (W_1=*"He"*, W_2 = *"buried"*, *, W_3 = *"life savings"*, *, *, *, *) might be associated with the ordered sequence of concepts: (C_1="**MISER**", C_2="**BURY**", *, C_3 = "**GOLD**", *,*,*,*) where the notation * is used to refer to a word (or word phrase) which is not assigned a concept for the purposes of coding the protocol data. The complex proposition F="**BURY(MISER, GOLD)**" would be assigned to the concept sequence (C_1="**MISER**", C_2="**BURY**", *, C_3 = "**GOLD**", *,*,*,*).

Once the assignments have been made, statistics are computed. Specifically, the probability that one concept follows another given a particular complex proposition (e.g., P(**BURY**|**MISER**, **BURY(MISER,GOLD)**) is estimated from the observed relative frequencies. In addition, the probability of a word given a concept is estimated (e.g., P(*"life savings"*| **GOLD**)). The probability that a given complex proposition is used is also estimated from the coder's behavior (e.g., P(**BURY(MISER,GOLD)**). Instead of assigning a zero probability to transition and emission probabilities whose corresponding observed relative frequencies were equal to zero, a small "smoothing" probability was used to facilitate processing of novel word sequences. Figure 2 shows a possible HMM representation for the complex proposition **BURY(MISER,GOLD)**.

Protocol Data Coding Algorithm

The Viterbi algorithm (Viterbi, 1967) as described in Allen (1995, p. 202) was then used to construct the "most probable" concept sequence associated with each possible complex proposition for a particular text segment. The "information content" in bits (i.e., a normalized log-likelihood measure) I of a complex proposition F consisting of M concepts C_1, C_2, ..., C_M and represented by M word phrases W_1, ..., W_M is computed using the formula:

where *log[x]* denotes the logarithm base 2.

$$I = -(1/M)\log\left[p(F)\prod_{i=1}^{M} p(C_i \mid C_{i-1}, F)\,p(W_i \mid C_i)\right]$$

Next, the complex proposition which was "most probable" (i.e., had the smallest information content score I) was selected. Complex propositions whose information content exceeded some maximum critical value were discarded and those text segments were defined as "incomprehensible" to AUTOCODER. This threshold was set sufficiently high, however, so that the occurrence of "incomprehensible" complex propositions was very rare. Notice that unlike the usual HMM approach to syntactic and semantic parsing, a unique HMM is constructed for each complex proposition rather than trying to construct a general HMM applicable to all possible complex propositions which could occur in the text.

Procedure

Three human coders jointly coded the recall data from the training data set using AUTOCODER. The human coders were careful not to examine the test data, so the dictionaries created as a result of coding the training data were likely to not contain all concepts, complex propositions, and statistics necessary to code the test data set. Text segments in the test data were then identified by the three human coders as well. AUTOCODER then assigned the "most probable" complex proposition to each text segment using the information content score described in the previous section. The three human coders then coded the test data without the use of AUTOCODER and measures of agreement between AUTOCODER's performance and the human coder performance on the test data set were recorded.

Results and Discussion

In order to compare performance of AUTOCODER and the human coder on the test data set, three different measures of agreement were used. All measures were computed individually for each text across all relevant subject data. It is important to emphasize that AUTOCODER always codes the same set of protocol data in exactly the same manner with 100% reliability. Thus, the agreement measures actually are measures of the validity as opposed to the reliability of AUTOCODER's coding performance.

Agreement Measures

The first measure was *percent agreement* which is defined as the percentage of times the two coders agree that a proposition was mentioned in the recall protocol plus the percentage of times the two coders agree that a proposition was not mentioned. One difficulty with the percent agreement measure is that percent agreement can be artificially increased by simply increasing the number of complex

propositions in the proposition dictionary! Accordingly, other agreement measures were considered.

The second measure of agreement was Cohen's Kappa score (Cohen, 1960) which essentially corrects for agreement by chance. The formula for Cohen's Kappa is given by: $\kappa=(p-p_c)/(1-p_c)$ where p is the *percent agreement* described in the previous paragraph and p_c is the expected agreement between the two coders if the coding strategy of one coder provided no information (i.e., was statistically independent of the coding strategy of the other coder). The performance of the model for the percent agreement and kappa agreement measures on the training data set is provided in Table 1. The quantity N denotes the number of opportunities for agreement. Typically, in the text comprehension literature. Percent agreement scores for coding data which are above 90% and kappa scores which are above 70% are deemed acceptable for publication. The data was also analyzed using a third more stringent agreement measure we call *sequential agreement*. Sequential agreement is typically not computed. But since the same coder has identified the text segments in both the training and test data, the percentage of times both the human coder and AUTOCODER agreed upon the coding of a particular text segment across recall protocols could be computed. This coding criterion thus takes into account the sequential structure of the recall data unlike the previously described agreement measures which are typically reported in the literature.

Analysis of Training Data

Table 1 shows the performance of AUTOCODER on the training data set using standard agreement measures, while Table 2 shows the performance of AUTOCODER using the sequential agreement measure. As can be seen from Tables 1 and 2, AUTOCODER's performance clearly demonstrates that it is picking up on a sufficient number of statistical regularities from the skilled human coder's data to almost completely reconstruct the skilled human coder's decisions.

Table 1: Performance of Autocoder on Training Data (Standard Agreement Measures)

Text	N	Percent Agreement	Cohen Kappa
"Miser"	192	95%	91%
"Cuckoo"	336	93%	84%
"Doctor"	228	99%	97%
"Eagle"	384	97%	93%

Table 2: Performance of Autocoder on Training Data (Sequential Agreement Measures)

Text	N	Percent Agreement
"Miser"	111	90%
"Cuckoo"	111	86%
"Doctor"	105	98%
"Eagle"	150	92%

Analysis of Test Data

Tables 3 and 4 show the performance of AUTOCODER on the test data set using the standard agreement measures and the sequential agreement measure. As can be seen from Tables 3 and 4, AUTOCODER's performance is almost comparable to experienced human coders keeping in mind the limitation that the test data set was parsed into text segments corresponding to complex propositions by a human coder. On the other hand, the AUTOCODER methodology has the important advantage that it is entirely well-documented and can be reliably implemented by computer software (unlike coding schemes implemented by human coders).

Table 3: Performance of Autocoder on Test Data (Standard Agreement Measures)

Text	N	Percent Agreement	Cohen Kappa
"Miser"	192	83%	65%
"Cuckoo"	336	88%	71%
"Doctor"	228	88%	75%
"Eagle"	384	84%	66%

Table 4: Performance of Autocoder on Test Data (Sequential Agreement Measures)

Text	N	Percent Agreement
"Miser"	111	69%
"Cuckoo"	111	67%
"Doctor"	105	76%
"Eagle"	150	68%

To provide a qualitative feeling regarding AUTOCODER's performance. Table 5 shows AUTOCODER's "coding" of the protocol data of Subject 12 who was assigned to the test data set.

It is extremely encouraging (despite the simple texts considered in this initial study) that the performance of the AUTOCODER algorithm was so effective on the test data. In almost all cases, AUTOCODER automatically and reliably coded the data at an almost publishable agreement level using completely documented and accessible algorithms. We are excited and pleased with these preliminary results even though the text segments in the test data had to be pre-

parsed by a human coder. Future work in this area is currently being pursued.

Table 5: AUTOCODER's "coding" of novel recall data

Human Recall Data	**AUTOCODER Interpretation**
"and he buried it in the ground"	**BURY** **AGENT:** MISER **OBJECT:** GOLD
"and he went over every day to look at where the money was where the lump of gold was buried"	**ATTEND** **AGENT:** MISER **OBJECT:** GOLD
"and one day when the miser wasn't there a thief came and dug up the gold"	**ATTEND** **AGENT:** MISER **OBJECT:** GOLD *[Disagrees with Human Coder!]*
"and so the miser goes and he sees the hole in the ground"	**BURY** **AGENT:** MISER **OBJECT:** GOLD *[Disagrees with Human Coder!]*
"and he's very upset by that"	**MISER** **STATE:** PLEASED *[Disagrees with Human Coder!]*
"and a bystander tells the miser to take a rock and bury it in the ground"	**TELLS-INFO** **FROM:** NEIGHBOR **TO:** MISER **INFO:**BURY(STONE)
"and the miser says why"	**ATTEND** **AGENT:** MISER **OBJECT:** GOLD *[Disagrees with Human Coder!]*
"and the bystander says well all you ever did was look at the ground anyway"	**TELLS-INFO** **FROM:** NEIGHBOR **TO:** MISER **INFO:**ATTEND (MISER,GROUND)
"you never did use the gold"	**TELLS-INFO** **FROM:** NEIGHBOR **TO:** MISER **INFO:**NOTUSE (MISER,GOLD)
"so there might as well be a rock there"	**TELLS-INFO** **FROM:** NEIGHBOR **TO:** MISER **INFO:**ASGOOD (STONE,GOLD)

References

Allen, J. (1995). *Natural language understanding.* Redwood City, CA: Benjamin/Cummings.

Charniak, E. (1993). *Statistical language learning.* Cambridge, MA: MIT.

Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and Psychological Measurement, 20,* 37-46.

Earwood, J. (2000). *AUTOCODER: An intelligent assistant for coding protocol data.* Psychology Program Senior Honors Thesis. School of Human Development. University of Texas at Dallas. Richardson, TX.

Ericsson, K. & Simon, H. (1993). *Protocol analysis: Verbal reports as data.* Cambridge, MA: MIT.

Golden, R. M. (1997). Causal network analysis validation using synthetic recall protocols. *Behavior Research Methods, Instruments, and Computers, 29,* 15-24.

Graesser, A. & Franklin, S. (1990). Quest: A cognitive model of question-answering. *Discourse Processes, 13,* 270-304.

Jelinek, F. (1997). *Statistical methods for speech recognition.* Cambridge, MA: MIT.

Kintsch, W. (1998). *Comprehension: A paradigm for cognition.* New York: Cambridge.

Trabasso, T. & Magliano, J. (1996). Conscious understanding during comprehension. *Discourse Processes, 21,* 255-287.

van den Broek, P. & Trabasso, T. (1986). Causal networks versus goal hierarchies in summarizing texts. *Discourse Processes, 9,* 1-15.

Viterbi, A. J. (1967). Error bounds for convolutional codes and an asymptotically optimal decoding algorithm. *IEEE Transactions on Information Theory, 13,* 260-269.

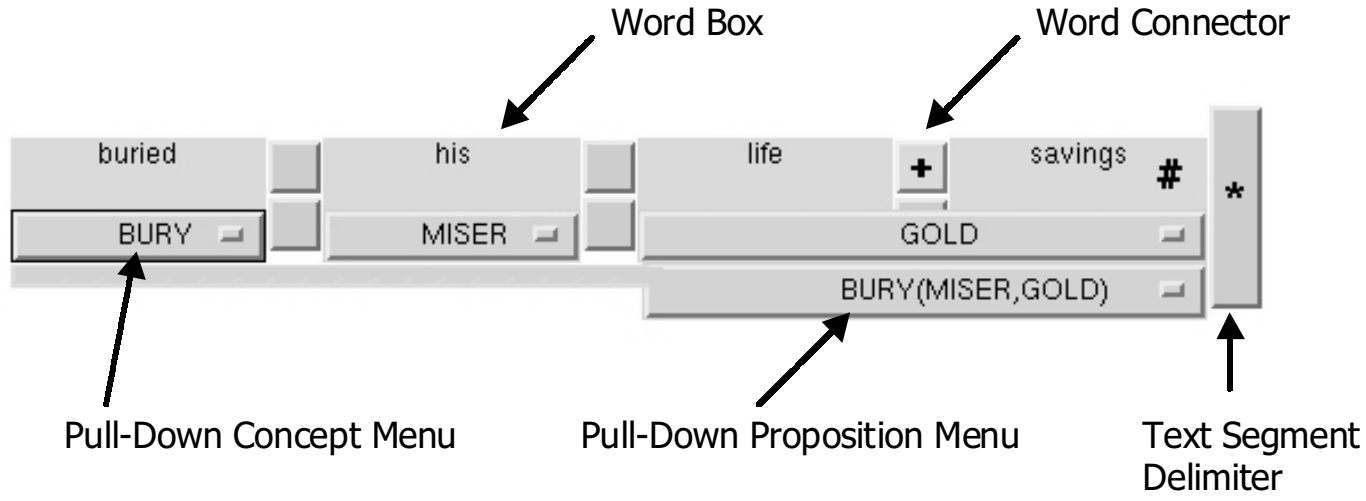

Figure 1. A portion of the AUTOCODER user-interface associated with the coding of the phrase *"buried his life savings"*. Each word in the text appears in a particular window called the *word box*. Word boxs can be connected to form word phrases using the *connector button*. Beneath each word phrase is a pull-down *concept menu*. Another pull-down *proposition menu* which lists the set of available complex propositions which can be assigned to the phrase is also displayed to the user. Both concept and proposition menus provide facilities for the addition of new concepts and propositions by the skilled human coder. Menu choices are made by a skilled human coder for the purposes of providing training data for the Hidden Markov Models (HMMs). The HMMs are then used to automatically make "most probable" menu selections without the aid of a skilled human coder through the use of the Viterbi algorithm for HMMs as described in the text.

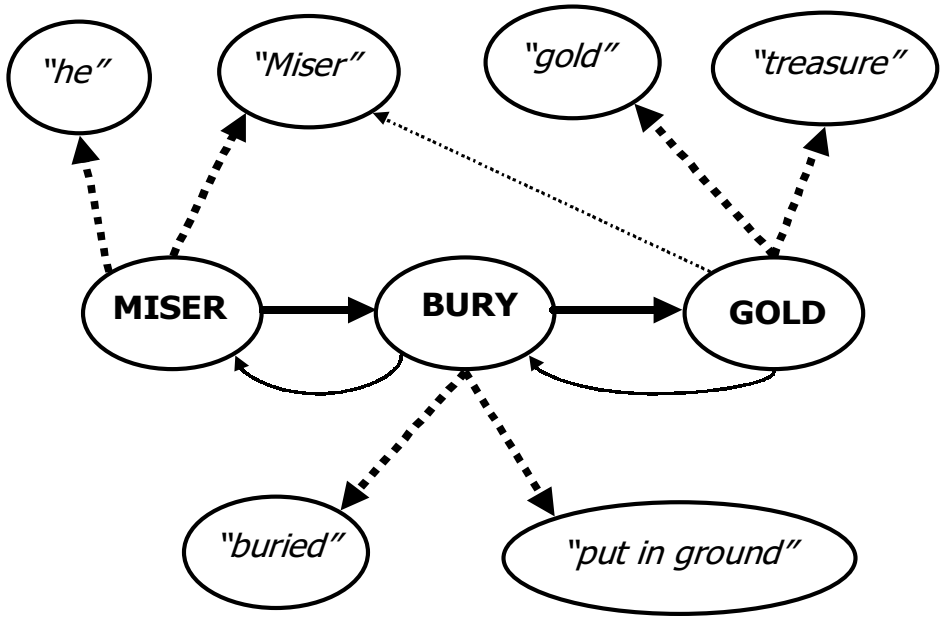

Figure 2. Each complex proposition is represented by its own HMM (Hidden Markov Model). In this figure, the HMM for the proposition **BURY(MISER, GOLD)** is graphically displayed. Transition probabilities are represented by solid arrows while emission probabilities are represented by dashed arrows. Line thickness indicates the relative magnitude of the corresponding transition or emission probability. Thus, the line thicknesses for the emission probability P(Word = "*gold*" | Concept = **GOLD**) and transition probability P(Concept=**GOLD** | Concept = **BURY**, Proposition = **BURY(MISER, GOLD)**) are both much thicker than the line thicknesses for the emission probability P(Word = "*Miser*" | Concept = **GOLD**) and transition probability P(Concept=**BURY** | Concept = **GOLD**, Proposition = **BURY(MISER, GOLD)**) .