International Journal of Semantic Computing

Corpus-Based Gesture Analysis: An extension of the FORM dataset for the automatic detection of phases in a gesture

Craig Martell
Naval Postgraduate School,
Monterey, CA, USA
cmartell@nps.edu

Joshua Kroll
Harvard University,
Cambridge, MA, USA
jkroll@hcs.harvard.edu

Received (16 January, 2008)
Revised (16 January, 2008)
Accepted (16 January, 2008)

We present the results of using an extension of the FORM gesture dataset to predict the mid-level phenomenon of phase. We compare the results of human phase predition with automated prediction using machine-learning techniques. Specifically, we present the results of hidden-Markov model experiments using an extended version of the FORM data to predict phase labels. Additionally, we compare FORM to the currently most accepted method of data gathering in this field—motion capture—by comparing the predictive accuracy of the physical gesture models produced by FORM and motion captur for phase labeling.

Keywords: Gesture. Corpus-based techniques. Machine learning. Hidden-Markov Models

1. Introduction

The most obvious way that humans communicate is through speech. As such, there has been a great deal of work in Linguistics, Logic, and Computer Science aimed at understanding, formalizing, and automatically generating and analyzing all aspects of human speech. However, speech is not the only means of communication available to us; we are able to send complex and subtle messages to each other via a variety of other means as well.

Gesture, for example, is an important channel for conveying intent and meaning. However, the field of gesture studies has only very coarse-level categorizations covering the types of gestures and very little in the way of fine-grained description techniques. Gestures are commonly divided into only four broad categories—beat, iconic, metaphoric, and deictic—and each gesture is further decomposable into its constituent phases. These phases are essentially of only four types: preparation,

stroke, hold and retraction[a]. However, the need for a more fine-grained system is well understood. In [12], Wittenburg et al., when describing the choices they made while designing their annotation scheme, state that "it was soon perceived that an exhaustive gesture encoding including all relevant characteristics would be ideal but impossible (except for small segments)."

At least some gestures are classifiable as alluded to above. Further, these gestures may be able to be broken down into their constituent phases. However, a coding (or annotation) scheme that only labels the gesture as a whole runs the risk of missing important variations in meaning created by subtle changes in the components of a gesture in question. Similar to facial expression, slight differences in the make-up of a beat gesture, for example, may well express very different things concerning the mood or intention of the speaker.

Accordingly, [7] developed a fine-grained, gesture coding scheme—FORM—that allowed annotators to exhaustively capture the constituent physical parts of the gestures of video-recorded speakers. In this article, we present an extension of FORM and the results of using it to predict the mid-level phenomenon of phase. In particular, we compare the results of human phase predition with automated prediction using machine-learning techniques. Specifically, we present the results of hidden-Markov model experiments using the extended FORM data to predict these phase labels. Additionally, we compare extended FORM to the currently most accepted method of data gathering in this field—motion capture—by comparing the predictive accuracy of the physical gesture models produced by FORM and motion capture for phase labeling.

2. The FORM Annotation Scheme

FORM is designed as a series of tracks representing different aspects of the gestural space. Each track is a series of gesture features (here referred to as "objects") parameterized by time. Generally, each independently moved part of the body has two tracks, one track for Location/Shape/Orientation, and one for Movement. When a part of the body is held without movement with respect to the gesturer, a Location object describes its position and spans the amount of time the position is held. When a part of the body is in motion, Location objects of zero duration are placed at the beginning and end of the movement to show where the gesture began and ended. Location objects with zero duration are also used to indicate the Location information at critical points (maxima, minima, and points of inflection) in certain gestures, i.e., location objects occur at keyframes.

An object in a movement track spans the time period in which the body part in question is in motion. It is often the case that one part of the body will remain static while others move. For example, a single hand shape may be held throughout a gesture in which the upper arm moves. FORM's multi-track system allows such

[a]Cf. [2], [3], [5], [6], and [8]

disparate parts of single gestures to be easily annotated separately. Once all tracks are filled with the appropriate information, it is easy to see the structure of a gesture broken down into its anatomical components.

2.1. *Extending FORM by Adding a* $5 \times 5 \times 5$ *Grid to FORM*

For the experiments described in this paper, we have extended FORM to include additional attributes and values for wrist location. These allow us to specify in a $5 \times 5 \times 5$ grid the x, y, and z coordinates of the wrist (or *end-effector*). This grid is relative to the location of the body, thereby eliminating the problem of incidental motion. To accomplish this, we simply take the sternum to be (2,2,2).

2.2. *The FORM Corpus*

FORM is a corpus of about 22 minutes of Brian MacWhinney teaching a Research Methods course at Carnegie Mellon University. These data were chosen since they were freely available via the TalkBank project (http://www.talkbank.org). They have been very useful for the project as people often gesture in a clear and exaggerated fashion while teaching.

3. Experimental Overview

As we have noted, we gestures can be segmented into the phases Preparation, Stroke, and Retraction with each adjacent phase possibly separated by a Hold. In this section, we will describe how we use the extended FORM representation to generate a matrix of vectors for each of these phases. We will then describe how we used this labeled data to run a series of hidden Markov model (HMM) experiments with the goal of *predicting* phase labels from the FORM representation.

As described in Section 2.1, we extended FORM by adding end-effector position given as $\langle x, y, z \rangle$ coordinates in a $5 \times 5 \times 5$ grid. If we combine these coordinates with the value of the *upperArmLift* parameter, we get a vector in $\mathbb{R}^4$ which describes the position of an arm at a particular frame. So, a sequence of these vectors encode the movement of an arm through-out a gesture excursion. If we divide the excursion into sub-sequences of these vectors such that they are co-extensive with the phase segmentation, we have created a set of labeled data.

However, FORM annotators only put *Location* markers at critical points in the gesture. The goal was to approximate zero-crossings in the first and second derivatives. In order to create the requisite interpolated vectors, then, we take the $\mathbb{R}^4$ vectors for each *Location* point in the gesture excursion and utilize various interpolation methods to fill in the values for the intervening frames. This generates a large matrix in $\mathbb{R}^4$, the number of columns of which is determined by the number of frames—at 29.97045 fps—in the excursion. We then divide this large matrix in accordance with the phase segmentation to generate bins of matrices representing the different phase types.

For each of the various interpolation methods, we then run the hidden Markov model experiment described in Figure 1. It is a version of a cross-validation method known as *Leaving-one-out* ([9]). For each iteration of the experiment the training set is of size $N-1$, while one data point, i, is used as held-out testing data. This process is repeated N times so each data point gets left out once.[b] Our particular algorithm works as follows. Of the combined set of *all* phase matrices—which we will call *observations*—choose one, $observation_i$, at each iteration and remove it from the set of observations. Then, for each of the sets of phases Preparation, Stroke, Retraction, etc, generate a hidden Markov model representing that phase and train with all the samples for that phase only. Label $observation_i$ after the hidden Markov model, M, which maximizes $P(observation_i|M)$. If the label generated for $observation_i$ matches the actual label of $observation_i$, call it a match. Finally, return $observation_i$ to the set of observations. We do this for all i. Our total percentage of matches is computed as 100 $\times$ (total matched/total number of *observations*).

function Leave-One-Out($phaseSet_1, \ldots, phaseSet_n$) **returns** percentage of correct classifications
  inputs: $phaseSet_1, \ldots, phaseSet_n$: a list of sets of phase matrices,
    e.g. *prepSet, strokeSet, retractionSet*
  local variables: *observation*: the current held-out phase matrix
    M_i: the HMM for $phaseSet_i$, e.g. M_{stroke} is the HMM trained on *strokeSet*
    match: a counter, initially 0, indicating number of correct classifications
    percentMatch: a number $\in$ [0,100], indicating percentage of correct classifications
for each *observation* **in** $\{phaseSet_1 \cup \ldots \cup phaseSet_n\}$
  for each $phaseSet_i$ **in** $(phaseSet_1, \ldots, phaseSet_n)$
    $M_i \Leftarrow$ createHMM($phaseSet_i$)
    if *observation* $\in phaseSet_i$ **then**
      Train(M_i) **using** $phaseSet_i -$ *observation*
    else
      Train(M_i) **using** $phaseSet_i$

  PredictedLabel(*observation*) $\Leftarrow$ argmax$_i$ P(*observation* | M_i)
  if PredictedLabel(*observation*) = actualLabel(*observation*) **then**
    match++

percentMatch $\Leftarrow$ 100 * (*match*/total number of observations)
return *percentMatch*

Fig. 1. Leave-one-out Training Algorithm using One HMM per Phase

[b]It is important to note that this method has both advantages and disadvantages. An advantage is that it allows for exploration of how the model changes for any particular piece of data. In addition, it is useful for doing cross-validation when the total number of data points is low, as is the case with the current FORM dataset. On the other hand, Chen and Goodman ([1]) argue that using larger deleted chunks gives better results. Here, we consider our method as giving something like an upper bound, as it is dangerously close to testing on the training data. As the size of the FORM dataset grows, we should be able migrate to safer methods.

4. Baseline

4.1. *Call-all-*X

In all of the experiments presented in this article, the baseline used is Call-all-x. Actually, Call-all-x is a combination of multiple baselines—one per phase—that produces particularly conservative results. For each of the phases, x, in the experiment, we assumed an algorithm that labels all observations as x. For example, the Call-all-Prep baseline labels every observation as a preparation. Precision is calculated simply as the proportion of actual preparations in the dataset. Recall will always be 1. *Mutatis mutandis* for all other phases.

These baseline results are conservative since the recall score for each phase is 1, which will drive up the baseline f-score for each phase. The important point here is that the high recall is *not* at the expense of precision. If it it were, then the f-score, being simply a special case of the harmonic mean, would be lower.

Finally, please note that sometimes baseline numbers are different across experiments using the same data set. This is an artifact of the HMM system we used[c]. Depending on how a particular HMM was trained, testing does not always complete. That is, the test observation may contain too few frames for one or more of the testing HMMs to return a probability. In these cases, GT2K/HTK simply returns an answer of "too few frames" and the observation is not labeled. However, to compare the results of experiments with different baseline numbers, we can simply look at the percent difference from the baseline f-score a particular algorithm is at predicting the phases. This is what is reported in the $\pm$***Baseline*** column of Tables I-XI.

5. PSR vs PSRH vs PSRHU

Because phase segmentation is not always cognitively clear, we added an "Unsure" category so that annotators could mark those frames in the penumbra between phases. However, this category can confuse prediction results because it subsumes features of all other phase categories. Additionally, there was a particular confusion between the Hold and Unsure categories. One would expect, then, that running *Leaving-one-out* using all five categories would produce lower results than running it with the Unsure category left out. Further, one would expect that experiments using just Preparation, Stroke, and Retraction would produce the best results. As the experiments in Tables I-IV show, these predictions turn out to be the case. For all of these experiments, we interpolated using cubic splines and vector quantized using the k-means algorithm with $k = 1000$.

Table I gives the results of running *Leaving-one-out* using all phases—Preparation, Stroke, Retraction, Hold, *and* Unsure—as inputs. As expected here, the Unsure category did very poorly with respect to the baseline f-score, as did

[c]We used HTK ([13]) and the Georgia Tech Gesture Toolkit ([11]) to build and train the HMMs.

Stroke. The other categories did better than baseline or were essentially the same. The poor results are not surprising given Table II, the confusion matrix for this experiment. Both Unsure and Hold caused a lot of off-diagonal confusion.

Table I also gives the results of running the experiment without the Unsure category. Here we see that removing Unsure increases our results as expected. From Table III, though, we can see that Hold is still creating a large amount of confusion. Again, this is due to the inconsistency of the annotation. Training annotators better on how to deal with incidental movement, so that holds are more uniformly annotated, may help with this.

Finally, Table I gives the results of just using Preparation, Stroke and Retraction. Table IV shows that this experiment has the least amount of confusion, although preparations are still often labeled as "Stroke."

6. First Experiments

In this section, we present the results of our first experiments using FORM. These are all location-based experiments, which were conducted as described in Section 3. We are calling them location-based experiments, because each vector of the phase matrices represents an arm location for that frame. This section is divided into balanced and unbalanced experiments. The balanced experiments attempt to maximize precision and recall for all three phases, while unbalanced experiments try to maximize precision at the expense of recall, which is useful in some applications.

6.1. *Balanced Experiments*

The results of the balanced experiments are given in Table V. All of the experiments are versions of *Leaving-one-out* with the differences being in the interpolation and vector-quantization methods used. In all of these experiments, the *UpperArmLift* parameter is simply linearly interpolated.

- Fixed-Grid: In this experiment we first linearly interpolated between the points given in the data. Then, we vector quantized by labeling each vector within a cube of the $5 \times 5 \times 5$ grid, e.g, $\langle 1, 1, 1 \rangle$, by the name of that cube. To do this, we simply rounded the results of the linear interpolation. So, for example, $\langle 1.235, 1.45, 1.75 \rangle$ becomes $\langle 1, 1, 2 \rangle$.
- L500: For this experiment, we first linearly interpolated between the points given in FORM, and then vector quantized using the fast k-means algorithm given in [4].[d] In this case $k = 500$.
- L1000: This experiment was the same as above except $k = 1000$.

[d]All vector quantizing for these experiments was done using the Matlab code available at http://www.cse.ucsd.edu/users/elkan/fastkmeans.html. We found it to be orders of magnitude faster than standard k-means.

PREPARATION	**Precision**	**Recall**	**F-Score**	**±Baseline**
Call-all-Prep-PSRHU	0.15	1.00	0.26	
PSHRU	0.38	0.66	0.48	+85%
Call-all-Prep-PSRH	0.19	1.00	0.32	
PSRH	0.45	0.67	0.54	+69%
Call-all-Prep-PSR	0.27	1.00	0.43	
PSR	0.50	0.68	0.58	+35%
STROKE	**Precision**	**Recall**	**F-Score**	**±Baseline**
Call-all-Stroke-PSRHU	0.30	1.00	0.46	
PSRHU	0.46	0.49	0.47	+2.2%
Call-all-Stroke-PSRH	0.40	1.00	0.57	
PSRH	0.58	0.60	0.59	+3.5%
Call-all-Stroke-PSR	0.54	1.00	0.70	
PSR	0.80	0.61	0.69	-1.4%
RETRACTION	**Precision**	**Recall**	**F-Score**	**±Baseline**
Call-all-Retraction-PSRHU	0.11	1.00	0.20	
PSRHU	0.49	0.56	0.52	+160%
Call-all-Retraction-PSRH	0.14	1.00	0.25	
PSRH	0.65	0.69	0.67	+168%
Call-all-Retraction-PSR	0.19	1.00	0.32	
PSR	0.72	0.82	0.77	+141%
HOLD	**Precision**	**Recall**	**F-Score**	**±Baseline**
Call-all-Hold-PSRHU	0.20	1.00	0.33	
PSRHU	0.43	0.26	0.32	-3.3%
Call-all-Hold-PSRH	0.26	1.00	0.41	
PSRH	0.53	0.27	0.36	-12%
UNSURE	**Precision**	**Recall**	**F-Score**	**±Baseline**
Call-all-Unsure	0.25	1.00	0.40	
PSRHU	0.34	0.24	0.28	-30%

Table 1. Precision, Recall, and F-Score Results for *S1000* on the *Brian* Data Set using Preparation, Stroke, Retraction, Hold and Unsure

- S500: In this case, we first used *spline3()* function from Matlab 7.0 to generate cubic splines between the FORM points. We then vector quantized with $k = 500$.
- S1000: As above, but $k = 1000$.
- SnoVQ: In this case, we used *spline3()* to generate cubic splines, but utilized all the vectors as given. That is, we did not vector quantize.

Truth \ Prediction	Preparation	Stroke	Retraction	Hold	Unsure
Preparation	69	22	6	2	5
Stroke	47	102	6	17	35
Retraction	5	3	42	9	16
Hold	24	38	14	36	24
Unsure	36	59	18	19	41

Table 2. Confusion Matrix for S1000-PSRHU

Truth \ Prediction	Preparation	Stroke	Retraction	Hold
Preparation	70	26	6	2
Stroke	53	125	8	21
Retraction	8	5	52	10
Hold	25	60	14	37

Table 3. Confusion Matrix for S1000-PSRH

Truth \ Prediction	Preparation	Stroke	Retraction
Preparation	71	26	7
Stroke	63	127	17
Retraction	8	6	62

Table 4. Confusion Matrix for S1000-PSR

6.2. *Unbalanced Experiments*

The unbalanced experiments we did are as follows. The results are in Table VI. For all of them we used cubic-spline interpolation.

- S1000.50: This is the same as S1000, above, but we added a measure of uncertainty. If the difference between the log-probability of the most likely model and the second most likely model was greater than 50% of the difference between the most likely and the least likely, we deemed the labeling to be *uncertain*. The 50% mark was chosen empirically to give the best results. We did this in the hope of increasing our precision, even at the expense of recall. As is evident, we did raise precision, but only from 0.5 to 0.51, while recall dropped from .68 to .38.
- P.25.500: For this experiment, we explored adding context from the previous phase, in the hopes that more context would increase the results. The P in the title indicates that we *prepending* context from the prior phase to the

	Prep			
	Precision	**Recall**	**F-Score**	**±Baseline**
Call-all-Prep	0.27	1.00	0.43	
Fixed-Grid	0.45	0.63	0.53	+23%
L500	0.50	0.64	0.56	+30%
L1000	0.46	0.67	0.55	+28%
S500	0.45	0.62	0.52	+21%
S1000	0.50	0.68	0.58	+35%
SnoVQ	0.47	0.69	0.56	+30%
	Stroke			
	Precision	**Recall**	**F-Score**	**±Baseline**
Call-all-Stroke	0.54	1.00	0.70	
Fixed-Grid	0.78	0.60	0.68	-2.9%
L500	0.79	0.61	0.69	-1.4%
L1000	0.78	0.59	0.67	-4.3%
S500	0.78	0.58	0.67	-4.3%
S1000	0.80	0.61	0.69	-1.4%
SnoVQ	0.79	0.59	0.68	-2.9%
	Retraction			
	Precision	**Recall**	**F-Score**	**±Baseline**
Call-all-Retraction	0.19	1.00	0.32	
Fixed-Grid	0.76	0.83	0.79	+147%
L500	0.67	0.81	0.73	+128%
L1000	0.77	0.79	0.78	+144%
S500	0.70	0.80	0.75	+134%
S1000	0.72	0.82	0.77	+140%
SnoVQ	0.77	0.81	0.79	+147%

Table 5. Precision, Recall, and F-Score Results for balanced HMM Methods Using the *Brian* Data Set

current phase. The *25* indicates that we are prepending 25% of the prior phase. The 500 indicates that we vector quantized with $k = 500$.

- P.25.1000: As above, but $k = 1000$.
- P.25.1000.373: As above, but if the difference between the highest log-prob and the second-highest log-prob was .373 or greater of the difference between the highest log-prob and the lowest log-prob, we called the labeling *uncertain*. Again, this number was empirically chosen to maximize precision.
- A.25.500: In this experiment, we explored *appending* 25% of the following phase to the current phase. The idea is to use context from later in the gesture. Again, vector quantization was done with $k = 500$.

- A.25.500.148: As before, we appended 25% of the following phase to the current phase; and, if the difference between the highest and second highest log-probability was .148 or greater of the difference between the highest and lowest log-probability, the labeling was deemed *unsure*. As before, .148 was determined empirically to maximize precision.
- A.25.1000: Appended 25% of the following phase and vector quantized with $k = 1000$.
- A.25.1000.35: Same as above, but the difference cut off was empirically set to .35.

6.3. *First Results*

It is interesting to note that retractions seemed the easiest to classify, with its highest F-score being .79 for the Fixed-Grid, balanced method. If the stroke is the most definite aspect of the gesture, we would have expected it to be the easiest to classify. Retractions do have the easily identifiable characteristic of ending in a rest position. As well, there is usually only one per excursion; there may be many preparations and strokes per excursion. Additionally, although the first preparation of an excursion starts from a rest position, the subsequent ones do not. We were able to use the *difference-cut-off* technique in the unbalanced section to increase the precision of stroke recognition, but it was at great expense to recall. A.25.500.148 allowed for a precision of .92, but a recall of .18. Although this may not be useful for an automatic phase detector, it could be very useful for scientific exploration of strokes. It would give high assurance that those phases we automatically identified as strokes were actually strokes. One more thing of note here is that, although we were able to use the segmentation "tricks" described above to increase certain statistics, the best overall was simply using cubic-splines and $k = 1000$. Further, this was not that much better than the simplest, Fixed-Grid method that simply linearly interpolates and rounds the x, y, and z coordinates to the nearest integer value.

6.4. *Significance: McNemar's Test*

Of all the experiments above, S1000 performed the best overall. However, the important question of whether or not the results are significant needs to be addressed. Given that all of the above experiments labeled the same pieces of data, and given that there is not a strong difference among the results of the various experiments, McNemar's test, interpreted as a sign test, is very useful.

There was no significant between the different methods except for between S1000 and S500, with two-tailed p-value .49.

	Prep			
	Precision	**Recall**	**F-Score**	**±Baseline**
Call-all-Prep	0.27	1.00	0.43	
S1000.50	0.51	0.38	0.44	+2.3%
P.25.500	0.44	0.41	0.42	-2.3%
P.25.1000	0.35	0.25	0.29	-33%
P.25.1000.373	0.48	0.29	0.36	-16%
A.25.500	0.49	0.60	0.54	+26%
A.25.500.148	0.50	0.41	0.45	+4.7%
A.25.1000	0.45	0.49	0.47	+9.3%
A.25.1000.35	0.49	0.29	0.36	-16.3%
	Stroke			
	Precision	**Recall**	**F-Score**	**±Baseline**
Call-all-Stroke	0.54	1.00	0.70	
S1000.50	0.92	0.06	0.11	-84%
P.25.500	0.74	0.65	0.69	-1.4%
P.25.1000	0.69	0.73	0.71	+1.4%
P.25.1000.373	0.90	0.18	0.30	-57%
A.25.500	0.74	0.59	0.66	-5.7%
A.25.500.148	0.92	0.18	0.30	-57%
A.25.1000	0.72	0.57	0.64	-8.6%
A.25.1000.35	0.83	0.20	0.32	-54.3%
	Retraction			
	Precision	**Recall**	**F-Score**	**±Baseline**
Call-all-Retraction	0.19	1.00	0.32	
S1000.50	0.87	0.61	0.72	+125%
P.25.500	0.15	0.56	0.24	-25%
P.25.1000	0.50	0.66	0.57	+78%
P.25.1000.373	0.54	0.71	0.61	+90%
A.25.500	0.30	0.69	0.42	+31%
A.25.500.148	0.30	0.69	0.42	+31%
A.25.1000	0.49	0.80	0.61	+91%
A.25.1000.35	0.53	0.72	0.61	+91%

Table 6. Precision, Recall, and F-Score Results for unbalanced HMM Methods Using *Brian* Data Set

7. Dual Data: FORM vs Motion Capture

The results described above may or may not be of interest depending on how well FORM compares to more precise ways of gathering gestural data. Additionally, the experiments above only concern one subject, Brian MacWhinney. In order to

address these issues we built another data set for comparison: the *Craig* data set. It comprises approximately three minutes of Craig Martell in a lecturing mode discussing his teaching methods. The data for this set were gathered in two ways: motion-captured and video-recorded. The video recordings were than annotated using extended FORM. This data set, then, allows us to compare FORM to motion-capture *vis-á-vis* prediction of preparations, strokes, and retractions. It also allows us to compare the prediction results of two FORM datasets of different speakers in similar situations.

Table VII gives the results of these experiments. Again, note that the baselines may be different among different experiments.[e] They are described below in order of their listing in the table.

7.1. *Location-Based Experiments*

Our first set of experiments on this dataset were, again, location-based. These were as follows.

- S1000-Craig: This experiment is the same as the original S1000 experiment for *Brian*[f]. The frames between the location points were interpolated using cubic splines and then vector quantized to 1000 vectors.
- SnoVQ-Craig: This experiment is the same as the original SnoVQ for *Brian*. It is just as above but without the vector quantization.
- mocap1000: For this experiment, we utilized a subset of the 32 motion-capture marker points to generate end-effector position for each arm, as well as *upperArmLift*. This created vectors in $\mathbb{R}^4$ analogous to those used in FORM. We then vector quantized to 1000.
- mocapNoVq: Same as above but without vector quantizing.
- simulatedFORM: In this experiment, we generated a set of key frames in the motion-capture data, performed cubic-spline interpolation, and vector quantized to 1000. The key-frames were chosen so as to match the location points given by the FORM annotation. This experiment was done to see if the location information of FORM combined with the fidelity of motion-capture could increase results.

7.1.1. *Results*

The first result of note is that for *Craig*, both FORM-annotated and motion-captured, the SnoVQ experiment did better than the S1000 experiment. This is likely due to the fact that the *Craig* data set has only three minutes worth of gesturing, as opposed to 20 minutes worth for the *Brian* data set. In the later case, the total number of *raw* vectors, so to speak, is much greater than with the *Craig*

[e] Cf. Section 4.
[f] The original FORM dataset

	Prep			
	Precision	**Recall**	**F-Score**	**±Baseline**
Call-all-Prep	0.35	1.00	0.52	
S1000-Craig	0.65	0.50	0.57	+5.8%
SnoVQ-Craig	0.67	0.50	0.57	+5.8%
mocap1000	0.56	0.45	0.50	-3.8%
mocapNoVQ	0.61	0.49	0.54	+3.8%
simulatedFORM	0.37	0.91	0.53	+1.9%
	Stroke			
	Precision	**Recall**	**F-Score**	**±Baseline**
Call-all-Stroke	0.45	1.00	0.62	
S1000-Craig	0.72	0.69	0.70	+13%
SnoVQ-Craig	0.72	0.73	0.72	+16%
mocap1000	0.64	0.53	0.58	-6.5%
mocapNoVQ	0.69	0.60	0.64	+3.22%
simulatedFORM	0.25	0.01	0.02	-9.7%
	Retraction			
	Precision	**Recall**	**F-Score**	**±Baseline**
Call-all-Retraction	0.20	1.00	0.33	
S1000-Craig	0.58	0.86	0.69	+109%
SnoVQ-Craig	0.61	0.86	0.71	+115%
mocap1000	0.46	0.78	0.58	+76%
mocapNoVQ	0.46	0.76	0.57	+73%
simulatedFORM	0.41	0.22	0.29	-12%

Table 7. Precision, Recall, and F-Score Results for Various HMM Methods Using the *Craig* Data Set

data. This would account for the benefits gained from using vector quantization, since, as the number of training vectors increases, the number of singleton vectors also increases. Additionally, Brian MacWhinney is teaching using a white board. This results in more movements away from the solar plexus than for Craig Martell, who is only addressing an audience. Vector quantization is useful under these circumstances as it creates equivalence classes of vectors with a representative vector for each.

The second interesting result is that FORM did better than motion-capture in predicting phase labels. Look at SnoVQ-Craig vs mocapNoVQ. The former does 5.8% better than Call-all-Preparation, while mocapNoVQ only does 3.8% better. This is not a large difference, but now consider Call-all-Stroke. Here SnoVQ-Craig did 16% better than baseline, while mocapNoVQ only did 3.2% better. The largest difference is for Call-all-Retraction. There the gain over baseline was 115% and 73% respectively. Table VIII gives the result of the McNemar's test for SnoVQ-Craig vs

Prep	**Correct**	**Error**	**p-Value**
Correct	35	16	1.0
Error	15	35	
Stroke	**Correct**	**Error**	**p-Value**
Correct	57	31	0.09
Error	18	22	
Retraction	**Correct**	**Error**	**p-Value**
Correct	41	10	0.18
Error	4	4	
All	**Correct**	**Error**	**p-Value**
Correct	133	57	0.05
Error	37	61	

Table 8. mocapNoVQ (along the top) vs. SnoVQ-Craig (down the side)

mocapNoVQ. Again, it is the case that although there is not a significant difference for any one phase, the difference overall is significant.

These results are contrary to expectation. One would think that the "physical truth" given by motion-capture would do better at predicting a mid-level *physical* phenomenon like phase than would a cognitition-laden annotation scheme. These results, then, may indicate that phase prediction is more cognitively based than was originally thought. One possible explanation for this is that FORM smoothes incidental movement. For example, there may be some subset of the parameters of a phase that humans use for classification, and the smoothing of the curve done by the FORM method may better approximate these parameters. Much more work needs to be done here, but incidental movement is a known difficulty for gesture and phase prediction.[g]

Simulating FORM by using the location points given by annotators to pick out key-frames in the motion captured data, however, did not work very well. If the above theory is correct, though, it should have. That is, if the reason for FORM's doing better than motion-capture is simply the smoothing of the curve and the removal of incidental movement, then simulatedFORM should have achieved this. A possible answer is that the FORM key-frames smooth further by picking out very *chunky* locations in space. The motion-capture time-stamp at some form location frame, i, will pick out a much more precise region of space. The FORM method reduces a large number of paths from $location_i$ to $location_{i+1}$ to just one, further reducing potential sparse-data problems.

[g]Cf. [10] for pointers.

	Prep			
	Precision	**Recall**	**F-Score**	**±Baseline**
Call-all-Prep-Lec	0.42	1.00	0.59	
S1000-Paul-Lec	0.67	0.61	0.64	+8.5%
Call-all-Prep-Con	0.45	1.00	0.62	
S1000-Paul-Con	0.63	0.65	0.64	+3.2%
	Stroke			
	Precision	**Recall**	**F-Score**	**±Baseline**
Call-all-Stroke-Lec	0.44	1.00	0.61	
S1000-Paul-Lec	0.70	0.62	0.66	+8.2%
Call-all-Stroke-Con	0.43	1.00	0.60	
S1000-Paul-Con	0.64	0.58	0.61	+1.7%
	Retraction			
	Precision	**Recall**	**F-Score**	**±Baseline**
Call-all-Retraction-Lec	0.13	1.00	0.23	
S1000-Paul-Lec	0.37	0.64	0.47	+104%
Call-all-Retraction-Con	0.12	1.00	0.21	
S1000-Paul-Con	0.56	0.68	0.61	+190%

Table 9. Precision, Recall, and F-Score Results S1000 for *Paul-Lecture* Data Set

8. More Data: *Paul*

The experiments in the prior sections have primarily cut across two dimensions. We first looked at FORM location-based experiments and then compared them to motion-captured location-based experiments. Secondly, we compared location-based experiments to motion-based experiments. In this section we will look at two more comparisons: FORM location-based experiments for multiple-subjects in an analogous contexts and FORM location-based experiments for the same subject in different contexts. We do this to check that changes in subjects or changes in contexts do not radically affect our results.

In order to do these experiments, we first created a third FORM data set, *Paul*. It contains roughly six minutes of Paul Howard; for the first three Paul is lecturing, for the second three Paul is having a conversation with someone off camera. S1000 was run using each of these subsections of the data. The results are given in Table IX.

8.0.2. *Results*

We see that they follow the same pattern for all three subjects. Retractions are the easiest to classify, followed by preparations and, finally, strokes. Although strokes for Brian were below the baseline, the difference is very small. This can, again, be accounted for by the fact that Brian was lecturing to a class and, therefore, using

the white board quite a bit. This increased both the location and shape ranges of his strokes. The most interesting thing here is that the 20 minutes of data in *Brian* didn't do as well as the 3 minutes of *Craig* or the 3 minutes of *Paul*. Brian's preparations may be cleaner than either Paul's or Craig's. Craig's strokes were the most easily picked out and, therefore, may also be cleaner. Further research should be conducted to discover the characteristic of the cleanest members of each phase. We should then explore whether these prototypes can be used in learning algorithms. For example, we may label a phase based on which prototype it is closest to.

Table IX also compares the results of the two sets of *Paul* experiments. *Paul/Conversational* didn't do as well at predicting preparations or strokes as did the *Paul/Lecturing*. This could simply be caused by the fact that Paul's gestures were, subjectively judged, more exaggerated when he was lecturing than in conversation. Retractions were predicted better from the conversational data.

9. Summary

In this paper, we presented an extension to the FORM annotation scheme for gathering gestural data along with a series of experiments designed to serve as a verification. Although, at this point, these results are preliminary, we have seen that FORM can be used to predict the mid-level annotation of PSR theory at least as well as the baseline when using cubic splines to interpolate. And, depending on the person and the amount of data collected, it may sometimes be better to vector quantize and sometimes not. Furthermore, these results hold for the cross-context experiment as well. Finally, it turns out that motion-capture does not do as well at predicting phases as does FORM, nor do motion-based experiments do as well as location-based ones. There is still too little data at this point to generate a full theory. However, these first results look promising.

References

[1] Stanley F. Chen and Joshua Goodman. An empirical study of smoothing techniques fo language modeling. *ACL*, 34:310–318, 1996.

[2] David Efron. *Gesture and Environment*. King's Crown Press, Morningside Heights, NY, 1941.

[3] David Efron. *Gesture, Race, and Culture: A Tentative Study of Some of the Spatio-Temporal and "Linguistic" Aspects of the Gestural Behavior of Eastern Jews and Southern Italians in New York City, Living Under Similar as well as Different Environmental Conditions*. Mouton, The Hague, 1972.

[4] Charles Elkam. Using the triangle inequality to accelerate k-means. In *Proceedings of the Twentieth International Conference on Machine Learning (ICML'03)*, 2003.

[5] Adam Kendon. Some relationships between body motion and speech: An analysis of an example. In Aron Wolfe Siegman and Benjamin Pope, editors, *Studies in Dyadic Communication*, pages 177–210. Pergamon Press, 1972.

[6] Adam Kendon. How gestures can become like words. In Fernando Poyatos, editor, *Cross-Cultural Perspectives in Nonverbal Communication*. C. J. Hogrefe, Toronto, 1988.

[7] Craig Martell. *FORM: An Experiment in the Annotation of the Kinematics of Gesture*. PhD thesis, University of Pennsylvania, 2005.
[8] David McNeill. *Hand and Mind: What Gestures Reveal about Thought*. University of Chicago Press, Chicago, 1992.
[9] Hermann Ney, Sven Martin, and Frank Vessel. Statistical language modeling using leaving-one-out. In Steve Young and Gerrit Bloothooft, editors, *Corpus-Based Methods in Language and Speech Processing*, pages 174–207. Kluwer Academic, Dordrecht, 1997.
[10] Francis Quek et al. Gestural origo and loci-transitions in natural discourse segmentation. Technical Report VISLab-01-12, Department of Computer Science and Engineering, Wright State University, 2001. http://vislab.cs.wright.edu/Publications/QueBMH01.html.
[11] Tracy Westeyn, Helene Brashear, Amin Atrash, and Thad Starner. Georgia tech gesture toolkit: Supporting experiments in gesture recognition. In *International Conference on Perceptive and Multimodal User Interfaces*, 2003.
[12] Peter Wittenburg, Stephen C. Levinson, Sotaro Kita, and Hennie Brugman. Multimodal annotations in gesture and sign language studies. In *International Conference on Language Resources and Evaluation.* European Language Resources Association, 2002.
[13] S. Young. The htk hidden markov model toolkit: Design and philosophy, 1993.