

**IMPROVING THE EFFICACY OF AUTOMATED SIGN LANGUAGE  
PRACTICE TOOLS**

A Thesis  
Presented to  
The Academic Faculty

by

Helene M. Brashear

In Partial Fulfillment  
of the Requirements for the Degree  
Doctor of Philosophy in the  
School of Interactive Computing,  
College of Computing

Georgia Institute of Technology  
August 2010

Copyright © 2010 by Helene M. Brashear

# IMPROVING THE EFFICACY OF AUTOMATED SIGN LANGUAGE PRACTICE TOOLS

Approved by:

Dr. Thad Starner, Advisor  
School of Interactive Computing,  
College of Computing  
*Georgia Institute of Technology*

Dr. Gregory Abowd  
School of Interactive Computing,  
College of Computing  
*Georgia Institute of Technology*

Dr. Vicki Hanson  
School of Computing,  
College of Art, Science and Engineering  
*University of Dundee*

Dr. Charles Isbell  
School of Interactive Computing,  
College of Computing  
*Georgia Institute of Technology*

Dr. Beth Mynatt  
School of Interactive Computing,  
College of Computing  
*Georgia Institute of Technology*

Date Approved: July 5, 2010

*To Infinty!*

*And Beyond!*

*I would like to thank those who have contributed their timee to making science magical for kids. Ernest brought me rocks and talked to me seriously about the science fair when I was a very little girl. Theresa let me “play” LOGO in the computer lab after school. Wanda Claire encouraged me on the path for the Girl Scout badge in mathematics when I was just a little Brownie. Mr. Wizard gave me a thousand ways to drive my mom crazy. Dr. Palmer had a big Santa Claus laugh and a mad scientist laboratory of delights for a classroom. Ms. Bensen was “Coach” for Math Team and inspired us to medals. Dr. White was “Coach” for Computer Team and hosted Saturday morning computer team and hamburgers for both a varsity and junior varsity team. Dr. Sinclair’s dedication and enthusiasm for the official Nerd Farm of Texas is both delightful and inspiring.*

*I would like to thank my parents for believing in their little engineer and her geeky ways. I would like to thank my family for driving me all over creation for math team, computer team, science fair, and all the other activities. My friends and family have been generous and giving beyong belief in their support for finishing this research and graduating.*

*Finally, I would like to thank my best friend and husband Brad. His patience with these shenanigans has been remarkable. He has been my rock, in all things.*

## ACKNOWLEDGEMENTS

This work is funded, in part, by the National Science Foundation and the National Institute on Disability and Rehabilitation Research. This material is based upon work supported by the National Science Foundation (NSF) under Grant No. 0511900. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of NSF. The Rehabilitation Engineering Research Center for Wireless Technologies is sponsored by the National Institute on Disability and Rehabilitation Research (NIDRR) of the U.S. Department of Education under grant number H133E060061. The opinions contained in this paper are those of the author and do not necessarily reflect those of the U.S. Department of Education or NIDRR.

Additional financial support has also been provided to me during my time at Georgia Tech by the Georgia Tech College of Computing, the GVU Center, and Google.

# TABLE OF CONTENTS

DEDICATION . . . . .	iii
ACKNOWLEDGEMENTS . . . . .	iv
LIST OF TABLES . . . . .	xi
LIST OF FIGURES . . . . .	xiv
SUMMARY . . . . .	xvii
I INTRODUCTION . . . . .	1
1.1 CopyCat . . . . .	1
1.2 Thesis Statement . . . . .	2
II RELATED WORK . . . . .	5
2.1 Sign Language Background . . . . .	5
2.1.1 Signed Languages . . . . .	5
2.1.2 American Sign Language . . . . .	6
2.1.3 Language Fluency . . . . .	7
2.1.4 SEE . . . . .	8
2.1.5 Signing Space . . . . .	9
2.1.6 Linguistics of American Sign Language . . . . .	12
2.1.7 Minimal Pairs in Signing . . . . .	14
2.1.8 Disfluencies in Signing . . . . .	15
2.1.9 Conversational Repairs . . . . .	16
2.2 Automatic Speech Recognition . . . . .	17
2.2.1 Speech Recognition Tasks and Corpora . . . . .	18
2.2.2 Speech Recognition with HMMs . . . . .	20
2.2.3 N-best Filtering and Re-scoring . . . . .	21
2.2.4 Evaluating Dialogue Systems . . . . .	21
2.3 Modeling Signed Languages . . . . .	23
2.3.1 Automatic Sign Language Recognition . . . . .	24
2.3.2 Sign Language Corpora . . . . .	25
2.3.3 Sensor Selection . . . . .	27

2.3.4	Recognition Techniques . . . . .	29
2.3.5	Basic Unit of Modeling . . . . .	31
2.3.6	Applications of Automatic Sign Language Recognition . . . . .	32
III	COPYCAT PROJECT . . . . .	35
3.1	Evolution of CopyCat System . . . . .	35
3.1.1	The CopyCat Games . . . . .	36
3.1.2	Language Learning . . . . .	36
3.1.3	Educational Evaluation . . . . .	37
3.2	System Design . . . . .	41
3.2.1	Interface Design . . . . .	41
3.2.2	Wizard of Oz . . . . .	41
3.2.3	Sensors . . . . .	42
3.3	Resulting Corpus . . . . .	44
3.3.1	Overview of Data Collected . . . . .	44
3.3.2	Characterizing the Children’s Signing . . . . .	44
3.4	Challenges of the CopyCat Corpus . . . . .	46
3.4.1	Library Continuity . . . . .	46
3.4.2	Sensor Changes . . . . .	47
3.4.3	Varied Environments . . . . .	47
3.4.4	Data Integrity . . . . .	47
3.4.5	Automatic Annotation . . . . .	48
3.4.6	Maintaining the Library . . . . .	48
3.4.7	Sign Variation . . . . .	49
3.4.8	Influencing the Children’s Signing . . . . .	49
3.4.9	Privacy Issues . . . . .	49
3.5	Summary . . . . .	50
IV	LABELING . . . . .	51
4.1	Research Goals . . . . .	52
4.2	Data Set . . . . .	53
4.3	Criteria . . . . .	56

4.3.1	Developing the Criteria . . . . .	56
4.3.2	Criteria: Sign Label . . . . .	58
4.3.3	Criteria: Handedness . . . . .	60
4.3.4	Criteria: Quality . . . . .	63
4.4	Labeling Tool . . . . .	66
4.4.1	Design . . . . .	66
4.4.2	Data Labeler . . . . .	67
4.5	Labeled Sets . . . . .	70
4.6	Analysis of Labeled Data . . . . .	71
4.6.1	Unique Classes Per Set . . . . .	71
4.6.2	Distribution of Signs Across Students . . . . .	71
4.6.3	Dominant Hand Switching . . . . .	75
4.6.4	Quality of Signing . . . . .	76
4.7	Structure of Signed Examples . . . . .	80
4.8	Summary . . . . .	81
V	MODELING . . . . .	82
5.1	Data Processing . . . . .	82
5.1.1	Accelerometers . . . . .	82
5.1.2	Image Processing . . . . .	83
5.1.3	Hand Tracking . . . . .	84
5.1.4	Head Tracking . . . . .	87
5.1.5	Feature Vectors . . . . .	87
5.2	Recognition Experiments . . . . .	89
5.2.1	Testing on Training . . . . .	89
5.2.2	Leave-one-out . . . . .	89
5.2.3	Data Divisions . . . . .	89
5.2.4	Recognition Metrics . . . . .	90
5.3	Recognition Infrastructure . . . . .	91
5.3.1	Hidden Markov Models . . . . .	91
5.3.2	HTK . . . . .	93
5.3.3	GT <sup>2</sup> k . . . . .	94

5.3.4	GART . . . . .	94
5.3.5	Building the Infrastructure . . . . .	94
5.4	Summary . . . . .	95
VI	EXPERIMENTS . . . . .	96
6.1	Research Goals . . . . .	96
6.2	Leave-one-out Tests . . . . .	97
6.2.1	Experimental Sets . . . . .	97
6.2.2	Results . . . . .	98
6.3	Testing on Training . . . . .	100
6.3.1	Results . . . . .	100
6.3.2	Analysis of Chance . . . . .	100
6.3.3	Comparison to Previous Work . . . . .	101
6.3.4	Analysis of Handedness . . . . .	103
6.3.5	Analysis of Quality . . . . .	103
6.4	Training Models with GOOD Samples . . . . .	103
6.4.1	Hybrid Experiments . . . . .	104
6.4.2	Results . . . . .	105
6.5	Recognition of Game Vocabulary . . . . .	106
6.5.1	Error Analysis: Substitution Errors . . . . .	106
6.5.2	Error Analysis: Quality . . . . .	109
6.5.3	Error Analysis: Handedness . . . . .	110
6.5.4	Error Analysis: Non-sign gestures . . . . .	110
6.6	Analysis of Class Variance . . . . .	111
6.6.1	Class Structure . . . . .	111
6.6.2	Class Variance . . . . .	112
6.6.3	Results . . . . .	112
6.7	Post-processing to Support Game Play . . . . .	113
6.7.1	Parsing for ASL Signs . . . . .	113
6.7.2	Results . . . . .	113
6.8	N-Best Recognition . . . . .	114
6.8.1	Experiments . . . . .	115

6.8.2	Results . . . . .	115
6.9	Summary . . . . .	119
VII	AUTOMATING GAME PLAY . . . . .	120
7.1	Introduction . . . . .	120
7.2	Research Goals . . . . .	121
7.3	Characterizing the Human Wizard . . . . .	123
7.3.1	Wizard Behavior . . . . .	123
7.3.2	Evaluation Metrics . . . . .	124
7.3.3	Wizard Performance . . . . .	124
7.3.4	Bias of the Space . . . . .	125
7.4	Automating Game Responses . . . . .	125
7.4.1	Architecture . . . . .	125
7.4.2	Design . . . . .	127
7.4.3	Classifier . . . . .	127
7.4.4	Parser . . . . .	130
7.4.5	Baseline Performance . . . . .	131
7.5	Evaluation of the Automated Game . . . . .	133
7.5.1	Evaluation Metrics . . . . .	133
7.5.2	N-Best Comparison . . . . .	133
7.5.3	Automated Game Performance . . . . .	133
7.5.4	Comparison to Wizard . . . . .	138
7.6	Analysis . . . . .	141
7.6.1	Automatic Game Results . . . . .	141
7.6.2	Expanding N . . . . .	142
7.6.3	Extending Game Response . . . . .	142
7.7	Summary . . . . .	145
VIII	DISCUSSION AND FUTURE WORK . . . . .	146
8.1	Discussion . . . . .	146
8.1.1	Labeling Ontology . . . . .	146
8.1.2	Recognition Experiments . . . . .	147

8.1.3	Automatic Game . . . . .	148
8.2	CopyCat: Integrating a Quality Assessment . . . . .	150
8.3	CopyCat: Improving Feedback for Teachers . . . . .	150
8.4	CopyCat: Expanding Game Functionality . . . . .	150
8.5	CopyCat: Improving the Automatic Game . . . . .	151
8.6	Discriminative Power of Models . . . . .	151
8.7	Segmentally Boosted HMMs . . . . .	152
8.8	Refinement of Ontology . . . . .	152
8.9	Modeling Signed Languages . . . . .	153
8.10	Final Remarks . . . . .	153
IX	CONCLUSION . . . . .	155
APPENDIX A	TOOL OVERVIEW . . . . .	157
APPENDIX B	LABEL FREQUENCIES IN DATA . . . . .	163
REFERENCES	. . . . .	182

## LIST OF TABLES

1	Table of data collected during the CopyCat project . . . . .	45
2	Game vocabulary . . . . .	45
3	Level 1 phrases used in the CopyCat game data. Signs in brackets are optional signs. . . . .	54
4	Level 2 phrases used in the CopyCat game data. Signs in brackets are optional signs. . . . .	55
5	Level 3 phrases used in the CopyCat game data . . . . .	56
6	A comparison of the original game vocabulary and labels that were added.	62
7	Breakdown of sign distribution by variations in handedness. Variations in some signs result in their placement in multiple categories. . . . .	65
8	Permutations of labeling information used in evaluations. . . . .	71
9	Distribution of unique classes per set . . . . .	72
10	Distribution of dominant hands in student signing: Percentages show the portion of signs performed as right hand dominant, left hand dominant, or both (symmetrical signs which do not have a dominant hand). . . . .	78
11	Breakdown of quality in student signing . . . . .	79
12	Analysis of the labeled data for game grammar matching. For the column Label: <i>Game</i> indicates that non-game vocabulary signs are ignored and <i>All</i> indicates that all signs are used for comparison. For the column Match: <i>Phrase</i> indicates a game correct transcription and <i>Grammar</i> indicates a transcript match to the generalized game grammar, but not specifically to the intended phrase for the game scenario. . . . .	81
13	Feature vector description . . . . .	88
14	Distribution of samples by user . . . . .	90
15	Definition of variables used to calculate recognition metrics . . . . .	91
16	Percentage word accuracy for per student, per scheme. . . . .	98
17	Percentage word correct for per student, per scheme. . . . .	99
18	Percentage sentence correct for per student, per scheme. . . . .	99
19	Testing on training results . . . . .	100
20	Chance of randomly choosing the right class assignment for a sign . . . . .	101

21	Testing on training results: Comparison with previous methods. Effects of game only vocabulary (Set #6) and full sign label set (Set #5) on auto-segmented labels. Effects of auto-segmentation (Set #5) and manually determined sign boundaries (Set #0). . . . .	101
22	Game grammar by Level: Words in brackets ( [ ] ) are considered optional. Level 1 requires a three-sign phrase, with both adjectives optional. Level 2 requires a four-sign phrase, with the first adjective optional. Level 3 requires a five-sign phrase, with all signs required. . . . .	102
23	Testing on training results: Effects of added handedness label . . . . .	103
24	Testing on training results: Effects of added quality label . . . . .	103
25	A listing of the number of substitution errors (a single entry in the confusion matrix) and the relative counts of those errors. Each instance of a substitution error (swapping label A for label B) is tallied by the confusion matrix. The table shows the number of unique instances for each count of a substitution. There are 296 instances of specific substitution errors that occurred once (first row) and one instance of a specific substitution error that occurred 32 times (last row). . . . .	107
26	A listing of the top substitution errors . . . . .	108
27	A listing of the all substitution errors between BLUE and GREEN . . . . .	109
28	Recognition errors for non-sign gestures. Substitution errors do not directly add to total because substitution errors between two non-sign gestures are double listed - once for each gesture. . . . .	111
29	Comparison of observation variances that increase or decrease in Set #1, Set #2, and Set #3 with respect to the parent Set #0 . . . . .	112
30	Comparison of testing on training results with the raw recognition output and the post-processing output. . . . .	114
31	Percentage word accuracies for the N-best experiments . . . . .	115
32	Percentage word correct for the N-best experiments . . . . .	115
33	Percentage sentence correct for the N-best experiments. . . . .	116
34	Percentage word accuracy for the N-best experiments with post-processing. . . . .	116
35	Percentage word correct for the N-best experiments with post-processing. . . . .	116
36	Percentage sentence correct for the N-best experiments with post-processing. . . . .	116
37	Time in microseconds for recognition of a single phrase. Tests were run on a AMD Athlon <sup>TM</sup> 64 Processor 3200+ 1000MHz with 1 Gig of RAM . . . . .	119
38	Profile of the Wizard's performance . . . . .	123
39	Game vocabulary . . . . .	128

40	Game grammar by Level: Words in brackets ( [ ] ) are considered optional. Level 1 requires a three-sign phrase, with both adjectives optional. Level 2 requires a four-sign phrase, with the first adjective optional. Level 3 requires a five-sign phrase, with all signs required. . . . .	128
41	Examples of the Classifier’s tagging output. The transcripts are mapped to the correct phrase and correct matches are tagged by their group. . . . .	129
42	Evaluation of automatic game performance on the hand-labeled transcripts	131
43	Percentage accuracies for evaluation of automatic game accuracy. High and low values are in bold . . . . .	134
44	Raw phrase counts for evaluation of automatic game evaluation. . . . .	135
45	Hit rates for evaluation of automatic game evaluation . . . . .	136
46	Percentage accuracies for comparison of automatic game to Wizard. High and low values are in bold . . . . .	138
47	Raw phrase counts for comparison of automatic game evaluation to Wizard.	139
48	Hit rates for comparison of automatic game to Wizard . . . . .	140
49	Summative comparison of results of Set #3 with N-best $N = 10$ . The table shows a comparison between the Wizard’s live performance and the ground truth language evaluation, comparison between the automatic game and the Wizard’s live performance, and comparison between the automatic game and the ground truth language evaluation. . . . .	141
50	Reference for HTK training tools . . . . .	157
51	Reference for HTK testing tools . . . . .	158
52	Tallies of the classes for Set #0. Shows the number of examples for each label, as well as the number of unique signers for that sign . . . . .	163
53	Tallies of the classes for Set #1. Shows the number of examples for each label, as well as the number of unique signers for that sign . . . . .	165
54	Tallies of the classes for Set #2. Shows the number of examples for each label, as well as the number of unique signers for that sign . . . . .	168
55	Tallies of the classes for Set #3. Shows the number of examples for each label, as well as the number of unique signers for that sign . . . . .	172
56	Tallies of the classes for Set #5. Shows the number of examples for each label, as well as the number of unique signers for that sign . . . . .	179
57	Tallies of the classes for Set #6. Shows the number of examples for each label, as well as the number of unique signers for that sign . . . . .	180

## LIST OF FIGURES

1	Signing space around the body (Image from Klima and Bellugi, “The Signs of Language” used with permission [73]) . . . . .	9
2	Signing space around the body (Image from Klima and Bellugi, “The Signs of Language” used with permission [73]) . . . . .	10
3	Signing regions on the body (Image from Klima and Bellugi, “The Signs of Language” used with permission [73]) . . . . .	10
4	Plane of movement: Signs with similar hand shape and movement that are differentiated by their positional plane. (Image from Klima and Bellugi, “The Signs of Language” used with permission [73]) . . . . .	11
5	Boisen and Bates’s evaluation architecture for dialogue systems (Image used from Boisen and Bates “A practical methodology for the evaluation of spoken language systems” with permission [15]) . . . . .	23
6	Example stills from the American Sign Language Linguistic Research Project at Boston University . . . . .	27
7	Polhemus trackers: A) Motion capture system for dance (Image used with permission from Polhemus, Inc.) B) Glove configuration used by Gao et al. (Image used from Fang, Gao, and Zhao “Large-Vocabulary Continuous Sign Language Recognition Based on Transition-Movement Models” with permission [39, 147, 40, 38] . . . . .	30
8	CopyCat screen shot from Mini Quests: A) Animated game characters in their worlds B) The villain (a snake) is hiding behind the chair. C) The push-to-sign button has a picture of the hero (Iris the cat) on it. Children push the button to sign to the hero and warn her about the snake. D) The live video feed allows children to see themselves as they sign. E) The help button has a picture of the sign for help and will cue ASL video to help the children during game play. F) This window displays the tutor videos. . . .	37
9	Results of expressive language tests during educational evaluations for the CopyCat Phase 2 deployments . . . . .	38
10	Results of the receptive language tests during educational evaluations for the CopyCat Phase 2 deployments . . . . .	39
11	Results of sentence repetition tests during educational evaluations for the CopyCat Phase 2 deployments . . . . .	40
12	Diagram of the Wizard of Oz system system showing A) live camera and sensor feed B) interface output split between wizard and user C) child’s mouse and D) the interface computer . . . . .	42
13	Gloves with accelerometer (top). Detail of wrist-mounted accelerometers (bottom). . . . .	43

14	Kiosk . . . . .	44
15	Snap shots of signs for CHAIR from two different children. The left example shows the ASL sign for chair, which uses two H-hands. The right example shows the SEE sign for chair, which uses an H-hand for the dominant (active) hand and a C-hand for the non-dominant (passive) hand. . . . .	59
16	Some examples of non-sign gestures. From left to right: sneeze, head scratch, chin pause, and WAVE. . . . .	60
17	START_SENTENCE (left) begins with the hand on the mouse, and then the hand moves into the signing space. END_SENTENCE (right) begins with the hand in the signing space, and then the hand moves to the mouse. . .	60
18	WAVE gesture from four different children. A) one-handed / right hand / hand facing away from body B) one-handed / left hand / hand facing away from body C) two-handed / hands facing away from body D) two-handed / hand facing towards body, . . . . .	61
19	CAT from three different children. The left examples show CAT performed right handed. The center example shows CAT performed with both hands (BOTH symmetric). The right example shows CAT performed left handed.	64
20	Screen shot of the Data Labeler with a labeled sample. . . . .	68
21	Close-up of the Data Labeler video clip editing controls. . . . .	69
22	Close-up of the Data Labeler label editing tabs and controls. . . . .	69
23	Close-up of the Data Labeler sample navigator. . . . .	69
24	Unique signers per class for Set #0 (Shown to visualize trends - not all labels are displayed) . . . . .	72
25	Unique signers per class for Set #1 (Shown to visualize trends - not all labels are displayed) . . . . .	73
26	Unique signers per class for Set #2 (Shown to visualize trends - not all labels are displayed) . . . . .	74
27	Unique signers per class for Set #3 (Shown to visualize trends - not all labels are displayed) . . . . .	75
28	Unique signers per class for Set #5 (Shown to visualize trends - not all labels are displayed) . . . . .	76
29	Unique signers per class for Set #6 (Shown to visualize trends - not all labels are displayed) . . . . .	77
30	Control system for the CopyCat game. The push-to-sign functionality segments the data. The control system synchronizes data streams from video and accelerometers, archives the raw data, and generates feature vectors for each signing segment. The control system also passes game information and the feature vectors to the automatic game, which returns a classification of the phrase as correct or incorrect. . . . .	83

31	Visualization of the FFT for a single three-axis accelerometer . . . . .	84
32	Hand segmentation process . . . . .	85
33	Segmentation of hands from video clips of the Phase One deployment . . .	86
34	Image of child signing which has been annotated with tracking information for the head and hands . . . . .	87
35	Screen shot from the verification tool (Image used courtesy of Zahoor Zafrulla)	88
36	Visualization of a four state left to right HMM with two skip states . . . . .	92
37	Mixture of Gaussians. From left to right: Single Gaussian, Mixture with two components, Mixture with three components . . . . .	92
38	Class re-mapping for the Hybrid experiments . . . . .	105
39	Comparison of the substitution errors and their frequencies. The chart dis- plays the data shown in Table 25 . . . . .	107
40	Charts for N-best recognition rates. The first row shows the recognition rates before post-processing. The second row shows the rates after post-processing.	118
41	Constructing the automatic game . . . . .	122
42	Diagram of data flow for automating the game . . . . .	126
43	State machine component of the Parser system. The state machine validates the transcription moving backwards through each entry. . . . .	132
44	Charts for the Automated Game Accuracy Rates, True Positive Rates, and True Negative Rates testing. The first row shows the same Y-axis scale for all graphs. The second row shows zoomed versions with local scaling for the Y-axis. . . . .	137
45	Trends for $N = 20$ experiments: Top graph shows trends for true positive, false negative, true negative, and false positive. Bottom graph shows trends for accuracy. . . . .	143
46	Extending the game responses by integrating the details on handedness and quality of signs (extension of the automatic game diagram in Figure 41) . .	144
47	GT <sup>2</sup> k interaction with application components (Image used from Westeyn, Brashear, Atrash, and Starner “Georgia Tech Gesture Toolkit: Supporting Experiments in Gesture Recognition” [151]) . . . . .	159
48	Data collection using GART (Image used from Lyons, Brashear, Westeyn, Kim and Starner “GART: The Gesture and Activity Recognition Toolkit” [89]) . . . . .	161
49	Recognition using GART (Image used from Lyons, Brashear, Westeyn, Kim and Starner “GART: The Gesture and Activity Recognition Toolkit” [89])	162

## SUMMARY

The CopyCat project is an interdisciplinary effort to create a set of computer-aided language learning tools for deaf children. The CopyCat games allow children to interact with characters using American Sign Language (ASL). Through Wizard of Oz pilot studies we have developed a set of games, shown their efficacy in improving young deaf children's language and memory skills, and collected a large corpus of signing examples. Our previous implementation of the automatic CopyCat games uses sign language verification in the infrastructure of a memory repetition and phrase verification task. Sign language verification compares each input phrase to the correct phrase and accepts or rejects the sample. This approach only accepts language usage which is an exact match for the game phrases.

The goal of my research is to expand the CopyCat system to use automatic sign language recognition and language processing to allow for more flexible language usage. I have created a labeling ontology from analysis of the CopyCat signing corpus, and I have used the ontology to describe the contents of the CopyCat data set. This ontology was used to improve automatic sign language recognition and to add a customized language processing component to the automatic game. Through these activities, I have created a automatic game component which combines automatic sign language recognition and language processing to enable dialogue-based interactions that better represent the usage of American Sign Language by the children in the CopyCat signing corpus.

# CHAPTER I

## INTRODUCTION

### *1.1 CopyCat*

Sign languages are used around the world by the deaf and speech impaired as a means of communication. These sign languages use hand, body and face gestures as well as spatial structures to communicate information.

Since early childhood is a critical period for language acquisition, early exposure to ASL is key for deaf children's linguistic development [93, 101]. Ninety-five percent of deaf children are born to hearing parents. Most of these parents do not know or are not fluent in sign language. Only 25% of parents of deaf children become fluent in ASL [97]. Often a child's first exposure to signing is at school. The slow development of language for these deaf children of hearing parents has been attributed to incomplete language models and interaction [55, 129]. The majority of deaf children of hearing parents remain significantly delayed in language development throughout their lives when compared with hearing children and deaf children of deaf parents [65, 127, 128]. In many cases they can be considered semi-lingual [54, 70] as they are fluent in neither English nor ASL.

For these deaf individuals, semilingualism is sometimes a life-long struggle [7]. Mayberry et al. [94] have shown that lack of early access to language interaction affects the child's adult level of language competence. This observation was true for both ASL and English. Deaf adults who received no access to sign interaction before age four were significantly less skilled in either language when compared to deaf adults who experienced sign language in infancy and deaf adults who lost their hearing after having acquired spoken English. The development of a language is dependent upon the availability of that language and the opportunities a child [78] or an adult learner [75, 76] have for interacting with skilled users of the language. Deaf children of hearing parents typically grow up in linguistically impoverished surroundings due to the inability of family members to use sign [48, 49]. The

quantity and quality of adult-child language interaction at an early age has also been shown to affect the language development and subsequent school success of hearing children [111].

In this dissertation I describe the development of CopyCat, which is a research prototype combining an interactive computer game with sign language recognition technology. CopyCat aims to assist young deaf children’s language acquisition by interactive tutoring and real-time evaluation. The goal is to encourage the linguistic transition from single, isolated utterances to phrase level signing. Unlike many educational games relying on English grammar skills or spoken audio files, CopyCat provides an English-free interface.

The gesture recognition developed as part of this dissertation work supports ASL-based communication between the user and the computer game. The child is asked to sit in front of the computer which is equipped with a video camera for the computer vision recognition system. He or she wears colored gloves with wrist-mounted accelerometers to assist the recognition. While playing the game, the child communicates with an animated character through ASL. This game is both mentally and physically engaging and allows the child to practice ASL in an enjoyable way.

Our previous implementation of the CopyCat system uses automatic sign language recognition and verification in the infrastructure of a memory repetition and phrase verification task [163, 164]. When children play the game, each game encounter has a scenario that must be described. The current system uses a phrase verification system, similar to those used in reading tutorial programs [4, 5], that verifies spoken input against an expected transcript from the reading passage. This verification approach only accepts language usage which is an exact match for the game phrase and produces a binary correct/incorrect format.

## ***1.2 Thesis Statement***

My automatic sign language recognition research is focused on creating user-independent models for recognition of hand-based American Sign Language gestures in order to expand the CopyCat game to a dialogue-based system. My data set is a collection of samples of the children interacting with the game via a Wizard of Oz setup. The recognition models include language from the game, vocabulary from out-of-game communication from the

children, and disfluencies discovered in the signing samples. I have designed a language processing component which allows for more flexible language usage including variations such as disfluencies, self-corrections, and non-sign activities. These changes expand the functionality of the CopyCat game, as well as the kind of feedback the game can provide.

In order to provide practical speech recognition applications, researchers studied and classified speech disfluencies [167]. Disfluencies that are commonly modeled in today’s recognizers include non-speech sounds such as coughing and sneezing as well as fillers such as “er” and “uh.” Since most of the sign language sets (including our own past work) used for machine learning have been collected in a controlled, laboratory environment, these sets do not fully explore common disfluencies in sign.

The transition from lab collected signing samples to real-world datasets for sign language recognition necessitates expanding models to include more diverse linguistic information. Just as the speech recognition community found that there is more to speech recognition than well-enunciated speech signals, there is more to sign language than perfectly performed signing. The linguistic scope of automatic sign recognition is still largely limited by technology, with most groups focusing on hand gesture recognition.

Our data set provides many samples of children signing casually as they interact with the online characters. It has many examples of non-signing activities such as scratching and fidgeting and also includes false starts, hesitations, and pauses. One of my goals is to explore the data and better characterize these disfluencies and valid variations in sign to aid in the pattern recognition task. This goal leads to my thesis statement:

**Thesis Statement:**

- Modeling the variations and disfluencies that occur in a casual signing context can improve the accuracy of a sign recognition system for an ASL practice tool and the performance of an automatic game.

In pursuing this thesis, my work makes several contributions to the field. First, I identify significant gestures in our dataset and create an ontology including game vocabulary, relevant non-game signs, communications directed towards game characters, and disfluencies in

sign (Chapter 4). I use this ontology to label 1191 phrases, resulting in 9055 segments representing a distinct sign, disfluency, or non-sign gesture. Of those labeled segments 48.03% were signs that were not contained in the game vocabulary (ex. IS, THE, WRONG) or disfluencies (ex. silences, fidgeting, hand waves, non-sign hand gestures). I labeled each sign as to its dominant hand: 79% were right-hand dominant, 17% left-hand dominant, and 4% were symmetric both. I labeled each sign for quality of performance: 94% were of GOOD quality, 4% were of OK quality, and 2% were of BAD quality.

The recognition system is described in Chapter 6. Baseline recognition results using previous methods show 60.78% and 61.19% word accuracies. I apply the my labelling ontology and show that the recognizer performs better (70.97% word accuracy) when each sign is modeled by several classes that distinguish variants of the sign based on quality and handedness. I further improve results to 77.22% by using post-processing for game play functionality. Exploring N-best recognition for multiple hypotheses results in a word accuracy increase to 87.10%.

The development of the language processing for the automatic game is described in Chapter 7. I then create a parser which maps a recognition transcript to whether a given utterance should be considered accepted as correct with respect to game play. This method resulted in a phrase accuracy of approximately 80%, which proved reasonable when compared to human performance in the task (92.85% phrase accuracy). Chapter 8 discusses future work, including research questions for the CopyCat project and more generalized automatic sign language recognition research questions. Chapter 9 summarizes and concludes the dissertation.

## CHAPTER II

### RELATED WORK

#### *2.1 Sign Language Background*

##### **2.1.1 Signed Languages**

Signed languages are used around the world by Deaf people for communication. Ethologue lists 130 different recognized sign languages globally [51]. Sign languages are not simply a pantomime of spoken languages but are their own independent, well-structured languages [138]. Additionally, there are not simple one-to-one correspondences to spoken languages. American Sign Language (ASL), British Sign Language (BSL), Irish Sign Language, Australian Sign Language (Auslan), and New Zealand Sign Language (NZSL) are all used in predominately English speaking countries [51]. Irish sign language is also referred to Deaf Sign Language in Ireland, this language has differing feminine and masculine language usage structure referred to as “women’s language” and “men’s language.” Traditionally women learned the masculine sign language when they begin dating [20, 51].

The difference in the evolution of spoken languages and their sign languages is indicative of a difference in Deaf and hearing cultures. Although ASL is used in a predominately English speaking country, ASL is in the same language family as French Sign Language (LSF), and not British Sign Language. This is due to the fact that the first school for the Deaf in the United States was founded by a Deaf French signer and not a Deaf British signer [107]

Here is a short summary of some of the structures in signed languages:

- **Finger spelling:** A subset of sign language which uses a manual alphabet to spell words. In ASL, these words are most commonly proper nouns from a spoken/written language. Loaner signs are a set of signs which symbolize basic concepts, but use finger spelling in a specific spatial pattern (usually circular) as a type of short cut.
- **Hand gestures:** The set of language gestures that use the hands to convey meaning.

They are generally described by hand shape, location, and movement.

- **Face gestures:** The set of language gestures that use the face to convey meaning. This set includes the use of eyebrows, mouth and facial expression, as well as head tilts. For example, eyebrows in ASL can be used to differentiate between classes of questions, and mouth gestures such as **cha** and **th** can differentiate meaning when used in conjunction with some hand gestures.
- **Body gestures:** The set of language gestures that use the body or torso to convey meaning. These gestures can include body shift for role change and body expansion or compression to add emphasis to hand or face gestures.
- **Inflections:** Some other gestures have variations in configuration that can affect their meaning either by emphasis, tone or adding context. These variations can include directional modifiers that indicate actor a gestures such as adding the “wiggle fingers” to a hand based gesture.
- **Spatial use:** Sign languages depend heavily on the use of space to convey meaning and provide context. Objects or persons can be placed in space and referenced later via indexing. Their placement in space can provide meaning such as over, under, next to, etc. Additionally, objects or persons can be referenced using classifiers which allow for a symbolic, spatial manipulation such as moving, activities like walking or dancing, or interactions such as collision or separation.

### 2.1.2 American Sign Language

American Sign Language (ASL) is a visual language with its own grammatical structure that uses hand, facial and body gestures to convey meaning [11]. ASL is the language used most frequently in face-to-face communication by the deaf. While the precise number of ASL users is difficult to determine, some estimates claim that ASL is the first or second language for between 250,000 and 500,000 Americans [2].

ASL was recognized as a language in the 1960s, and linguists have only recently begun to study sign languages in depth, starting in the late 1970s [86]. We have documents of

ASL starting with its roots in Old French Sign Language starting from the works of L'Epee in 1784 and can follow its migration to America with some signing manuals, including J. Schuyler Long's "The Sign Language: A Manual of Signs" printed in 1918. Films as early as 1913 document the movements of ASL and help us understand its evolution. Linguists have discussed the trends in ASL towards the signing bubble shown in Figure 1 and how signs change in shape and movement over time [73].

ASL, like any other language, is a living language that adapts and changes over time. New signs are added, and old signs are phased out. Signs such as CAT, COMPARE, and HORSE have evolved over time to be completely unrecognizable from their original sign [73]. ASL can be very flexible and adaptive in daily conversation. Signers continue a conversation even with interferences such as items in the hand, desks or tables occluding the lower body or hats on the head. Signers adapt their signing based on their partner; they will slow down or speed up, change vocabulary, or move along a continuum between ASL and Conceptually Accurate Signed English (CASE). All of these qualities are like any other language, humans adapt for understanding in communication [113].

### **2.1.3 Language Fluency**

For most native users of sign languages, spoken (or written) languages are usually a learned second language [138]. For native ASL signers English is a second learned language [106]. Ninety-five percent of deaf children are born to hearing parents. Most of these parents do not know or are not fluent in sign language [97]. Often a child's first consistent exposure to language is initially signing at school and later further emphasis on English language learning. Often a child's first exposure to signing is at school. The slow development of language for these deaf children of hearing parents has been attributed to incomplete language models and interaction [55, 129]. The majority of deaf children of hearing parents remain significantly delayed in language development throughout their lives when compared with hearing children and deaf children of deaf parents [65, 127, 128]. In many cases they can be considered semi-lingual [54, 70] as they are fluent in neither English nor ASL.

For these deaf individuals, semilingualism is sometimes a life-long struggle [7]. Mayberry

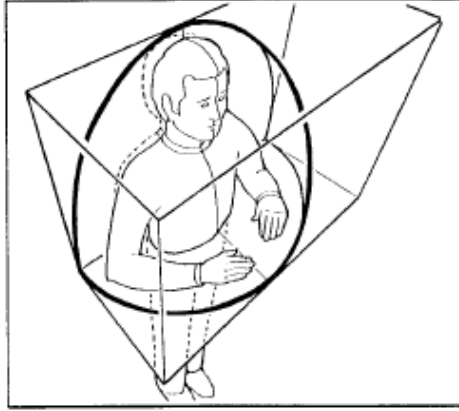
et al. [94] have shown that lack of early access to language interaction affects the child's adult level of language competence. This observation was true for both ASL and English. Deaf adults who received no access to sign interaction before age 4 were significantly less skilled in either language when compared to deaf adults who experienced sign language in infancy and deaf adults who lost their hearing after having acquired spoken English.

The development of a language is dependent upon the availability of that language and the opportunities a child [78] or an adult learner [75, 76] have for interacting with skilled users of the language. Deaf children of hearing parents typically grow up in linguistically impoverished surroundings due to the inability of family members to use sign [48, 49]. The quantity and quality of adult-child language interaction at an early age has also been shown to affect the language development and subsequent school success of hearing children [111].

Once children start school, the quality and consistency of their environmental language may vary substantially. A 2003 survey showed that at 47% of the sign based school programs, no more than half the instructors were classified as fluent in ASL [32]. Many educators practice simultaneous communication, which is an attempt to speak English and sign at the same time. The resulting sign typically has a high error rate and does not follow correct grammar practices of ASL [86]. These mixed language models directly impact children's ability to acquire first and second language skills [32, 86]. Although many Deaf people achieve a high level of proficiency in English, not all Deaf people can communicate well through written language; the average Deaf adult reads at approximately a fourth grade level [1, 63].

#### **2.1.4 SEE**

Another sign language that is used less frequently in the United States is SEE. What is commonly referred to as SEE is actually two systems: Seeing Essential English (SEE1) and Signing Exact English (SEE2). SEE2, which was created in 1972, is the most commonly used of the two [108]. Both SEE languages were designed by educators in response to second language learning problems most Deaf children faced when learning English and the often resulting low English fluency. The core philosophy in designing the SEE languages was that



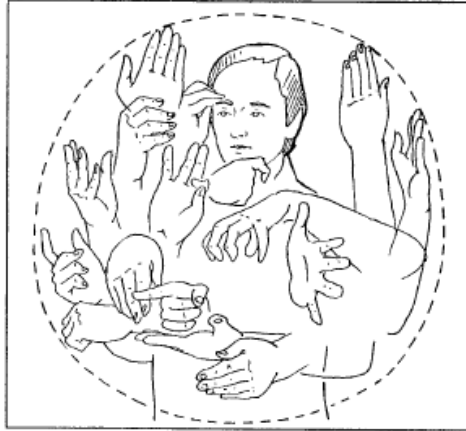
**Figure 1:** Signing space around the body (Image from Klima and Bellugi, “The Signs of Language” used with permission [73])

English should be spoken simultaneous to signing and that signs should have one exact meaning [108]. Many SEE signs are initialized version of ASL signs, where the hand shape is changed to modify the sign’s meaning.

### 2.1.5 Signing Space

In charades or pantomime people tend to use whole body movements that are restricted only by physiology. Large body movements are not uncommon, and the person pantomiming may use the space around them by walking, jumping or other such movements. The signing space for ASL is generally much more restrictive. The general signing space for ASL can be described as a bubble around the body. Figures 1 and 2 show a two dimensional and three dimensional diagram of the signing space around the body respectively. Most signs fall within this space, though there are some that fall outside. In the evolution of signs through history, signs tend to migrate away from the edges of the signing space towards the center or other signing planes [73].

Figure 2 shows a frontal, two dimensional view of the signing space. This figure shows that arm movement in the signing space is generally close to the body and does not allow for full extension of the arms. Most signs obey these constraints and are kept fairly close to the body. Figure 1 show the bubble of space around the body. This bubble is usually pictured from the waist to the top of the head. The elbows are usually kept close to the body, which



**Figure 2:** Signing space around the body (Image from Klima and Bellugi, “The Signs of Language” used with permission [73])

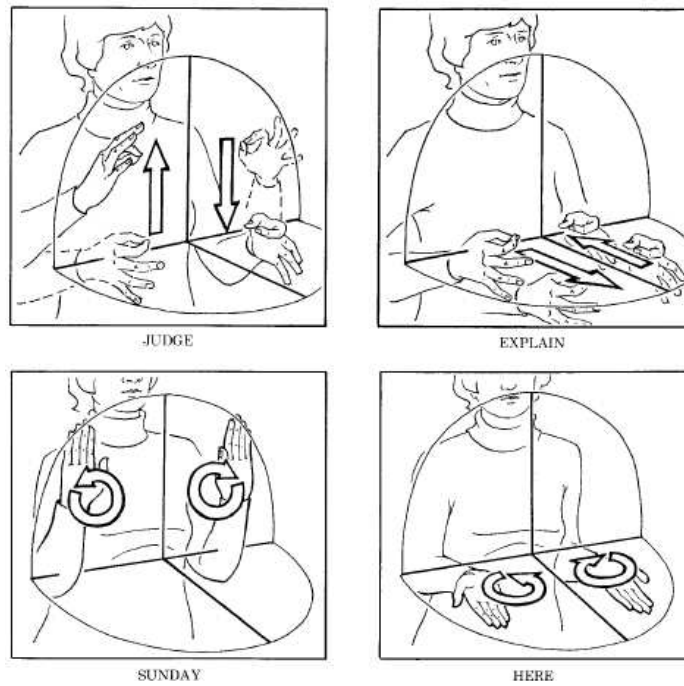


**Figure 3:** Signing regions on the body (Image from Klima and Bellugi, “The Signs of Language” used with permission [73])

helps define the sides of the signing bubble. The hollow of the neck is considered the center point of the signing bubble.

The general signing space can be divided into distinct places of articulation as seen in Figure 3 [73]. These locations on the body can be used to describe where a sign is coming from, going to, or near the space it moves through. This particular division of space leaves the signer’s chest as a single, open location. This space is considered to be divided into many intersecting planes that describe location. Figure 4 shows two examples of these planes and how they both define signs and help differentiate them from each other.

An interesting note is that violations of the signing space can be used for emphasis. Large signing and body movements can indicate a large item or great distance in space



**Figure 4:** Plane of movement: Signs with similar hand shape and movement that are differentiated by their positional plane. (Image from Klima and Bellugi, “The Signs of Language” used with permission [73])

or time. Sign also has the equivalent of whispering or yelling, which can be shown by size of movement, facial expression, and body movement. Whispering in sign is done by very closed, directed body movement in the direction of the recipient, and the signs are performed in a very small space.

Yelling can be done with exaggerated signs that use large movements. One observed example was a child who wanted his mother to tie his shoes. His father was trying to tie the shoes, but the whole time the child was crying and signing MOTHER and WANT. The child's signing used very large movements that went outside the normal signing space. MOTHER was extended outwards more than usual and the arms were almost entirely extended (which is normally unusual). WANT began outside the normal signing space and extended in very rapidly [113].

### **2.1.6 Linguistics of American Sign Language**

Stokoe demonstrated that ASL signs are not singular gestures, but are instead composed of components (phonemes) that are recombined to form a much larger lexicon [132, 120]. Since then linguistic researchers have done much work to further define the phonology and morphology of these components [138], though a large part of the focus has been on ASL, and it is only more recently that researchers have focused on cross-language sign linguistics universals [120, 121].

#### *2.1.6.1 Phonology*

Phonology refers to the study of the language structure below the word level. Phonology is the layer of language that is closest to the human production and perception layers; for speech these modalities are vocal and auditory, and for sign languages these are physical manual and visual [119, 120]. ASL signs are composed of five basic parts: hand shape, movement, location, palm orientation, and non-manual signs (facial expression) [138]. The Movement-Hold model defines signs as a series of sequential hold segments and movement segments. Holds are configurations that hold a steady state, and movements are transition times when some part of the sign changes.

Valli and Lucas use the sign GUESS to demonstrate the Movement-Hold model. The

sign is a HMM structure composed of hold (C-hand, location: right eye, orientation: palm up), movement, and hold (S-hand, location: left cheek, orientation: palm down). There are other possible sign structures including M, H, MH, MHMH, MMMH. Not every permutation of M and H are linguistically valid combinations [138].

#### *2.1.6.2 Morphology*

In spoken language, morphology refers to the study of the smallest meaningful components of words. In English an example would be to break the word “books” into the morphemes “book” and the plural suffix “-s”. Sign languages also have morphemes, which are the smallest meaningful components of signs. Liddell uses the sign AGREE as a demonstration of the morphemes of a compound sign [86]. The sign AGREE is composed of the signs for THINK and SAME-AS. The transition between THINK and SAME-AS are slightly modified versions of the end of the THINK sign and the beginning of the SAME-AS sign. Due to the structural changes that occur for sign transitions of compound signs and the historical changes that occur as languages evolve, a typical ASL compound usually consists of morphemes that are slightly modified [86].

These morphological compounds can be created by both concatenative operations and simultaneous performance [120]. The previous example of AGREE is a concatenative morphology, since it is a product of the sequential combination of two signs. The sign {TWO}{MONTH} is an example of a morphology created by simultaneous performance [86]. This sign uses the sign TWO and the sign MONTH together simultaneously, which results in a merging of the signs. The resulting sign uses the TWO sign V-hand to perform the actions for the sign MONTH that would normally be performed with a 1-hand.

These morphological changes can be characterized as movement epenthesis, hold deletion, metathesis, and assimilation [139]. Additional movements that are added as a transition between two phonemes are called the movement epenthesis. Hold deletions are the removal of a hold when two signs are combined sequentially. Metathesis is when the segments of a sign can change places. An example of this modification is when signs allow the signer to swap start and end locations while still retaining the meaning of the sign. When a segment

acquires the characteristics of another segment, such as in the sign {TWO}{MONTH}, it is called assimilation.

### 2.1.7 Minimal Pairs in Signing

In spoken languages, minimal pairs are words that differ by only one phoneme. The words *bat* and *pat* are minimal pairs, since they differ only by the starting phonemes of /b/ and /p/. Other examples from spoken English include *fan-van* and *site-side* [162].

The sequential nature of signing differentiates the idea of minimal pairs slightly from that of spoken languages. Minimal pairs in signed languages still are used to demonstrate a minimal phonological contrast [121]. They are structurally defined as a function of sequence, since the features that differentiate signs co-occur with other features of the sign simultaneously [138]. Signs can differ in hand shape, location, and movement, but these differences can also be distinguished by their sequential nature [120].

One example of a minimal pair is SISTER and BROTHER. The signs use the same hand shape and motion, but have differing starting positions. SISTER starts at the chin (where many feminine signs begin) and BROTHER starts at the forehead (where many masculine signs begin). Within the CopyCat game context there is one example of a minimal pair, GREEN and BLUE. These signs have the same location and movement, but differ by hand shape.

Valli and Lucas note that the simplicity of minimal pairs such as GREEN and BLUE can be a bit deceptive in light of the sequential nature of signing [139]. The example provided is the minimal pair THANK-YOU and BULLSHIT. These signs both start in the same configuration, with a B-hand at the chin and with the hand oriented palm towards the body. The movement of the signs is the same outward gesture and the ending location of the sign is the same position in front of the body. However the sign for BULLSHIT varies due to a transitional change in hand shape during the movement away from the chin. The sign THANK-YOU does not change hand shape. This example is used by the authors to illustrate the difficulties in annotating the morphological differences in minimal pairs.

### 2.1.8 Disfluencies in Signing

Most data sets for automatic sign language recognition are scripted data sets collected in the laboratory by the researchers [136]. Though scripted datasets provide a good test bed for developing research systems, the field of speech recognition has found that they are limited in their representation of how language is used and lack examples of common conversational artifacts such as accents (since they are often over-enunciated), disfluencies, and inflections [68]. On the other hand, speech recognition research has found that input to conversational interfaces typically contains disfluencies and out-of-vocabulary words [167]. This phenomenon has resulted in a body of research on modeling disfluencies for speech recognition and online word learning. Additionally conversational signing may contain register variation which may result in more or less formal signing depending on the signer's environment. Register variation is the relative level of formality (or informality) that is used by people in different situations. This register variation can affect how signs are performed, what vocabulary is chosen and what grammar is used [138]. Datasets collected in formal, scripted settings may lack many of the important language facets that are needed to fully model the language for use in live recognition systems.

Linguists have largely focused on the structure of sign languages and have done little work studying disfluencies [34].

Wallin suggests several disfluency categories: [34]

- Unfilled pauses: halted signs that pause and then continue
- Filled pauses: empty gestures
- Prolongations: signs extended by reiteration of loops
- Repetitions: repetition of part or all of the sign
- Restarts: repetition of the beginning hand shape or motion of a sign
- Truncations: incomplete signs
- Mispronunciations: signs executed incorrectly

Holt, et al [137] provide a comprehensive summary of the problems of automatic sign

language recognition research. Holt describes several significant problems specific to automatic sign language recognition: [136]

- Distinguishing gestures from signs
- Context dependency (directional verbs, inflections, etc)
- Basic unit of modeling (what are the phonemes? how do we describe them?)
- Transitions between signs (movement epenthesis vs. co-articulation effects)
- Repetition (cycles of movement in a sign may vary in length)

It is interesting to note that many of the significant problems of sign language relate to the list of issues with disfluencies. Distinguishing gestures from sign is a fundamental problem, since many of these non-sign gestures occur during conversation and act as disfluencies. Without modeling of the basic structure of signs, it is problematic to use traditional phonemic methods of modeling. Transitions between signs include both co-articulation effects and movement epenthesis. Co-articulation effects refer to the changing of signs when they overlap. Movement epenthesis is the actual movement between signs when there is a difference between a sign's ending location and the starting location of another sign. Distinguishing gestures from sign is a fundamental problem, since many of these non-sign gestures occur during conversation and act as disfluencies.

### **2.1.9 Conversational Repairs**

Sidnell defines conversation repair as “an organized set of practices through which participants in conversation are able to address and potentially resolve such problems of speaking, hearing, or understanding.” Repairs can be initiated by either the speaker or the conversational partner and are marked by disjunction of the conversational flow [125]. Self-repairs can be initiated by either the speaker (self-repairs) or the conversational partner (other-repairs) [122]. These kinds of repairs are also used during signing, and the cause of the repair is called the trouble source [27].

Of particular interest to us in the context of the CopyCat game are replacement repairs and word-search repairs. A replacement repair is a conversational repair that offers a replacement for the trouble source. Word-search repairs occur when a signer attempts to

recall a piece of information or is searching for the correct phrasing. Ethnographic studies have shown that word-search repairs occur more frequently than replacement repairs. Adult signers typically use non-manual features such as eye gaze or head movement to mark such repairs [27].

## *2.2 Automatic Speech Recognition*

Language recognition can be roughly divided into two parallel tracks: speech recognition and sign language recognition. The speech recognition track began in 1936 at AT&T's Bell Labs and has progressed to viable commercial applications. The sign language recognition track began in the early 90s and lags significantly behind speech recognition. There are a number of reasons for this gap, many of which trace to a lag in linguistic research between spoken and written languages versus signed languages. The formal study of signed languages can be dated to Stokoe's publication "Sign language structure: An outline of the visual communication systems of the American Deaf" in 1960 [35, 132]. Even with the publication of his work, Stokoe struggled to convince the research community that signed communications were truly languages.

Some key features that characterize these language recognition systems are:

- **Speaking mode:** isolated words versus continuous speech
- **Speaking style:** read speech versus spontaneous speech
- **Enrollment:** speaker dependent versus speaker independent
- **Vocabulary:** small (<10 words) to large (>20,000 words)

Additionally other criteria such as sound quality or language models can be used to characterize systems.

- **Proof of concept systems:** These early "proof of concept" systems were simple systems that showed that recognition could be done. They are characterized by small vocabularies, constrained environments, isolated utterances and user-dependant models.
- **User Independent:** These systems transitioned from single user system to language

models that would generalize to multiple users. The initial versions of these systems often only had a few different users from which to build models.

- **Large Scale Vocabulary:** These systems scaled vocabulary to a larger corpus. These larger vocabularies showed that the systems could be scaled to represent the scope of natural language vocabularies.
- **Continuous recognition:** These systems transitioned from isolated utterance recognition to continuous recognition of languages. These systems began to address co-articulation effects and grammatical structures for continuous language processing.
- **Large Scale User Base:** These systems transition from models built from a few users' data to large scale libraries of data from thousands of users. These corpora allowed for more generalized models and began to address many of the issues with accents, different voice patterns (as a result of gender, age, etc.), and regionalized vocabularies.
- **Task-based systems:** These systems constrain their vocabularies and grammar based on specific tasks. They are user-independent systems that use continuous recognition in a scripted, push-to-talk environment.
- **Transcription systems:** These systems allow for continuous language input and function as a live, transcription system. They are commercially available and built on a large scale vocabulary and user base.

### 2.2.1 Speech Recognition Tasks and Corpora

In the field of speech recognition there have been a series of tasks and corpora that have aided researchers by providing a point of reference for algorithms. Many of these tasks and corpora are used as baseline tasks to characterize progress in the field. A few of notable examples of these tasks and corpora are:

- **TIDIGITS (1984)** The Studio Quality Speaker-Independent Connected-Digit Corpus (TIDIGITS) was collected by Texas Instruments in 1984 for the purpose of “designing and evaluating algorithms for speaker-independent recognition of connected

digit sequences” [82]. The corpus contains data from 326 speakers reading digit sequences in both isolated and continuous sequences [83].

- **TIMIT (1986)** The DARPA TIMIT Acoustic-Phonetic Continuous Speech Corpus (TIMIT) was published in 1990 and contains a total of 6300 sentences, 10 sentences spoken by each of 630 speakers from 8 major dialect regions of the United States. The purpose of this database was to provide speech data for acoustic-phonetic studies, and it deliberately was chosen to represent multiple accents [46].
- **Switchboard (1990)** The Switchboard-1 Telephone Speech Corpus was collected by Texas Instruments in 1990 under DARPA program sponsorship. The corpus was collected under computer control and contains spontaneous, natural speech which represent every major dialect of American English. The complete set includes around 2340 conversations from over 500 speakers. These conversations were an average length of around 6 minutes and the corpus contains over 240 hours of recorded speech [47]. This corpus was collected for the purpose of providing a large multi-speaker database with extensive spontaneous (non-scripted) samples.
- **ATIS (1990)** The DARPA Air Travel Information System (ATIS) corpus was collected for research on spontaneous speech and natural language understanding. It was a task specific corpus generated by a Wizard of Oz system for booking flights. The data is divided into three sets: the first contains 912 spontaneous utterances from 36 speakers, the second contains 478 utterances read from the first set of transcripts, and the third contains 3171 utterance read by 10 speakers. The third set was collected specifically for the purpose of training speaker dependent speech recognition systems for the ATIS task [56, 77, 166].
- **TRAINS (1991, 1993, 1995, 1996)** TRAINS was a dialogue system for managing a railway transportation system. The system was designed to keep a queue of conversation acts, actively track the state of the agent’s goals and maintain the flow of conversation. The TRAINS corpus was initially collected in 1991 and expanded several times [68]. The scenario was constructed to mimic human to human interactions as a tool to design human computer interaction models for fluent conversations. The

corpus contains 98 dialogues from 34 speakers performing 20 different tasks [6].

## 2.2.2 Speech Recognition with HMMs

Hidden Markov models are used to model stochastic processes over time. The HMM  $\lambda = \{A, B, \Pi\}$  is defined by [110, 31]

- A set of **unobserved states**  $W^T = \{w_1, w_2, \dots, w_T\}$  that represent an underlying process
- A set of **transitional probabilities**  $P(w_j(t+1)|w_i(t)) = a_{ij}$  that represent the probability of transitioning to state  $w_j$  at time  $t+1$  given that we were in state  $w_i$  at time  $t$
- A set of **initial transitional probabilities**  $\pi = \{\pi_1, \pi_2, \dots, \pi_T\}$  that represent the initial probabilities of being in a state
- A set of **visible states**  $V^T = \{v_1, v_2, \dots, v_T\}$  that represent the observed state emissions
- A set of **observation probabilities**  $P(v_k(t)|w_j(t)) = b_j(k)$  that represent the probability of making an observation  $v_k$  while in state  $w_j$  at time  $t$

The three central problems of HMMs are then [110, 31]:

- **Evaluation Problem:** Given a fully trained HMM  $\lambda$  and a set of observations  $O$ , what is the probability that the observations were generated by the model:  $P(O|\lambda)$ ? The evaluation problem is commonly computed using the Forward-Backward procedure.
- **Decoding Problem:** Given a fully trained HMM  $\lambda$  and a set of observations  $O$ , what is the most likely sequence of hidden states  $w^T$  that generated the observations. The decoding problem is commonly computed using the Viterbi algorithm which finds the single best state path that maximizes  $P(Q|O, \lambda)$ .
- **Learning Problem:** Given an untrained HMM  $\lambda$  and a set of observations  $O$ , how do we adjust the parameter  $\lambda = \{A, B, \pi\}$  to maximize  $P(O|\lambda)$ . The learning problem is commonly computed using the Baum-Welch algorithm which uses expectation maximization to re-estimate the model parameters  $A$  and  $B$  alternately.

Hidden Markov models are popularly used in speech recognition and are used in almost all present day large vocabulary continuous speech recognition [43]. In the case of speech recognition, the observations are mapped to acoustic feature vectors, and each model represents a single phoneme [161, 43]. The decoding problem becomes the problem of trying to find the most likely word for a speech signal. A language model is usually applied to the HMMs to help model the combination of phonemes to words.

### **2.2.3 N-best Filtering and Re-scoring**

In the domain of speech recognition, an N-best list is a list of transcriptions and their associated alignments ranked by probability. N-best lists can be used as filters, where the lists are processed by a natural language system. The first example that is validated by the language filter is the one accepted. N-best lists can also be used with re-scoring, where the probability for each entry is combined with a score from the natural language processor. The new score is used to re-rank the list and choose the optimal transcription [112].

N-best lists have been used in speech recognition and have been especially successful in constrained tasks such as the ATIS task [77, 166]. Other domain tasks where N-best are frequently used include keyword spotting and phrase detection [68, 112]. N-best lists are also used as input to natural language processing systems such as in pen input systems like QuickSet [109].

### **2.2.4 Evaluating Dialogue Systems**

There are many methods of evaluating dialogue systems, depending on the project focus, domain, and specific application. King presents a survey of many of these techniques and the challenges they face in generalization [72]. Many of the early DARPA systems were task oriented projects designed to elicit improvements in specific fields of natural language processing and were focused on a series of domain specific metrics for evaluation [46, 56, 72]. The DARPA evaluation of these projects was based on successful completion of a task. However, internal evaluations by project managers were often based on metrics that check algorithmic performance for each component of the system for accuracy, computational time, and error diagnosis. These internal evaluations were often considered proprietary information, and

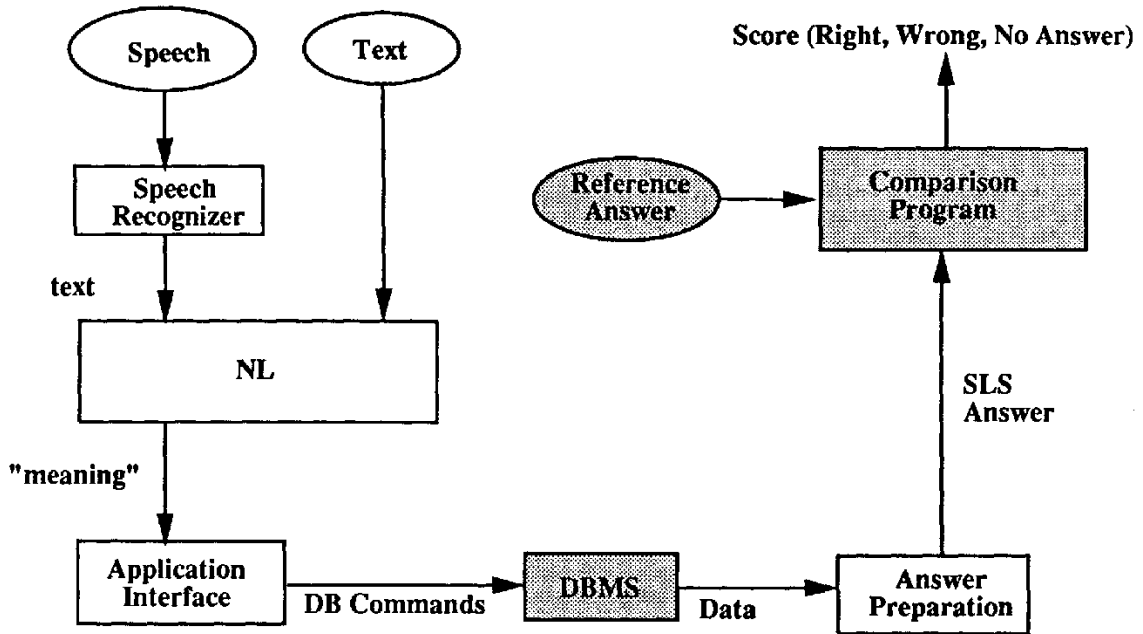
their techniques and results were not published at the time. This proprietary approach to internal development and evaluations is still a major factor in the field [72].

DARPA evaluation of the ATIS task was based on the system's end-result of the interaction. For ATIS, this end-result was the information retrieval from the database [66]. The ATIS project collected a series of queries and replies via both spoken and written English using the Wizard of Oz method. Transcripts of these interactions were used to generate models of user interaction with the dialogue systems. Teams working on the ATIS task used these transcripts in the development of their systems for evaluation of the recognition systems that they were building [77, 166]. The recognition systems were a key component of the ATIS system, but the final scoring of systems for each team was performed by a scoring of the correctness of the query / result pairing.

Boisen and Bates developed a methodology based on the collective experiences of BBN's participation in the DARPA projects [15]. Their methodology analyzed many domain specific evaluation methods to create a general framework to characterize the evaluation of dialogue systems. The Boisen and Bates methodology is a black box evaluation method, which means that it doesn't focus on the results of intermediary components, but is instead is focused strictly on the mapping of input to output. Figure 5 shows the authors' diagram of the evaluation architecture. This framework has been used in the design and evaluation of several of the DARPA challenge tasks, including the ATIS system.

Boisen and Bates define a series of principle elements:

- **Agreeing on Meaning:** There should be agreement on how queries are paired with answers. This pairing may be related to underlying architecture, such as a SQL database, or related to the definition of key terms or concepts related to the domain.
- **Reference Answers:** There should be a set which explicitly demonstrates the mapping between queries and answers. These answers are derived from the evaluation of linguistic principles and domain-specific criteria. For some tasks, it may be necessary to describe the answers as a continuum with some minimum answer and some larger, maximum answer. There is no specific restriction on how to generate reference answers, but Wizard of Oz testing is frequently used to collect the data for a reference



**Figure 5:** Boisen and Bates’s evaluation architecture for dialogue systems (Image used from Boisen and Bates “A practical methodology for the evaluation of spoken language systems” with permission [15])

set.

- **Comparison Software:** The comparison software is the test system that is being evaluated. This system should provide an answer to any test query. These answers will then be compared to the reference answers for scoring the system.
- **Scoring Answers:** There should be a numeric measure of the system for relevant criteria. These metrics should include information on significance and reliability of the metric. These scoring metrics provide a consistent measure of the evaluation across all test systems.

### 2.3 Modeling Signed Languages

Automatic sign language recognition is the process of using sensors to collect data from a user’s signing and using computers to recognize the signs. Though the focus of my research is on American Sign Language recognition, I will use the generic term of sign language in the literature review because there are many different sign languages. English speaking countries may use American Sign Language, Signed Exact English, Cued English, British

Sign Language, Irish Sign Language or Auslan, just to name a few. Many other language regions and countries have their own sign languages of varying etymologies. Research in automatic sign language spans the different sign languages across the globe.

There is often some confusion in the technology fields as to which sign language task is being researched. Here is a short summary of some technology based sign language research activities:

- **Automatic sign recognition** is the process of using sensors to collect data from a user's signing and recognizing the signs.
- **Sign generation** is a presentation of a visual representation of sign language, usually via a human avatar. These presentations use a representation (or gloss) of a sign language to generate a simulation of signing.
- **Written/Spoken language to sign translation** is a machine language translation task of converting written or spoken language into a representation (or gloss) of a sign language.
- **Sign to written/spoken language translation** is a machine language translation task of converting a representation (or gloss) of a sign language into a written or spoken language.

### 2.3.1 Automatic Sign Language Recognition

Sign language recognition is a growing research area in the field of gesture recognition. Research on sign language recognition has been done around the world, using many sign languages, including American Sign Language [17, 130, 146], Korean Sign Language [71], Taiwanese Sign Language [85], Chinese Sign Language [37, 44], Japanese Sign Language [115], and German Sign Language [13]. Many sign language recognition systems use Hidden Markov Models (HMMs) for their abilities to train useful models from limited and potentially noisy sensor data [44, 130, 146]. Sensor choices vary from data gloves [85] and other tracker systems to computer vision techniques using a single camera [130], multiple cameras, and motion capture systems [142] to hand crafted sensor networks [59].

Current state-of-the-art systems in the field of automatic sign language recognition (ASLR) can be characterized by 85-90% accuracy on data sets collected from one signer with about 100 words on average [137]. There are not many public data sets available, so most systems collect and use their own data. The field does not have the same set of baseline tasks and corpora that automatic speech recognition does.

Although most ASLR systems focus on a phoneme-based approach, they tend to limit the systems to a subset of phonemes which includes hand shape and location. Automatic speech recognition systems model phonemes and re-combine them in sequence to model words. As discussed in Section 2.1.6 of this chapter, sign language phonemes are not sequential in nature, but are more parallel dependent processes in nature. The Movement-Hold model explicitly describes this structure in terms of dividing a sign into sequential movement and holds [86]. Almost none of the research includes non-manual markers such as facial expression or body tilts [137].

Currently, there are no known systems that model context-dependant signs [137]. Context-dependent signs vary structure by their context. For example, many verbs in sign language can change structure depending on the subject and the predicate [120]. Likewise, positions in space can be used to represent people or concepts and later referred to for use as pronouns [138].

### **2.3.2 Sign Language Corpora**

Public data sets are often used to compare algorithmic performance in many fields, and speech recognition frequently uses public corpora to compare results. One major challenge of using these corpora for dialogue systems is that they are frequently narrow in their representation of language use [72]. Collecting, labeling, and distributing these corpora can be very time consuming and expensive. Examples of some popular speech corpora used to evaluate systems include the Brown corpus of English, the Trsor de la Langue Franais corpus of French, and the bilingual (English-French) parallel corpus drawn from the Canadian Hansard [72]. Public repositories of written information such as the “Wall Street Journal” have been frequently used for natural language processing tasks. The computer science

bibliography collection “Citeseer” lists 408 papers for the query “Wall Street Journal” [3].

Most data sets for automatic sign language recognition are scripted data sets collected in the laboratory by the researchers [136]. Though scripted datasets provide a good testing bed for developing research systems, the field of speech recognition has found that they are limited in their representation of how language is used and lack examples of common conversational artifacts such as accents (since they are often over-enunciated), disfluencies, and inflections [68]. Additionally conversational signing may contain register variation which may result in more or less formal signing depending on the signer’s environment. This register variation can affect how signs are performed, what vocabulary is chosen and what grammar is used [138]. Datasets collected in formal, scripted settings may lack many of the important language facets that are needed to fully model the language for use in live recognition systems.

The field of automatic sign language recognition has very few publicly available data sets, which makes it difficult to compare systems [137, 22]. Many data sets are used to recognize isolated signs, and in cases where the data sets have phrases, it is not always clear that the authors are referring to ASL phrases or simply a sequence of isolated signs put together [137]. The required labor to collect and label sign data is a major difficulty in the field of ASLR [23, 159, 105] . Recent collaborations between linguists and computer science researchers are beginning to result in publicly available corpora for use in the ASLR community.

One notable effort is the the American Sign Language Linguistic Research Project at Boston University. This project provides a public video data set that has been collected by linguists for use in research for sign linguistics [100], but has been used by researchers for ASLR [29, 99]. This project has also resulted in a collaboration between Boston University and the University of Texas at Arlington, which generated the “American Sign Language Lexicon Video Dataset.” This corpus contains approximately 3,800 signs, each of which is signed by between one and four native signers. A set of stills from the data set is shown in Figure 6 [10].

Researchers have taken several approaches to the problem of limited public data. Many



**Figure 6:** Example stills from the American Sign Language Linguistic Research Project at Boston University

have collected their own data sets in ASL [18, 59, 141], Chinese Sign Language [37, 44, 71], German Sign Language [13, 52], Auslan [62, 67, 69], and Japanese Sign Language [61]. Other researchers have focused on techniques for use with small data sets [22, 36].

Recent work in semi-autonomous data labeling has also been applied to ASL sets, to help reduce the labor involved in collecting data. Yang et. al use a two-pass hand segmentation method which uses human input in the second pass to aid in semi-supervised labeling of ASL data sets [159]. Cooper and Bowden explored semi-autonomous labeling on public signing videos of news broadcasts and successfully mined correlations between signs and captions with a word spotting rate of 53.7% for 23 signs that occurred four or more times during the broadcast [23].

### 2.3.3 Sensor Selection

Sensors used for ASLR can be divided into the following categories: cameras and gloves. A variety of cameras and computer vision techniques are used for various aspects of ASLR (see Section 2.3.3.1 for more information). Data gloves usually consist of sensors that measure flex and acceleration (see Section 2.3.3.2 for more information).

#### 2.3.3.1 Computer Vision

As noted in the previous section on corpora, most of the publicly available data sets are video based. Computer vision is frequently used in ASLR [13, 36, 60, 130, 145, 153]. Many systems use gloves and/or long sleeves to aid in tracking the hands [13, 36, 60, 64, 130, 156]

or require the user to return to a neutral position between signs [152, 165]. Hand tracking and characterization have been a central theme of most ASLR research [105, 137]. Many researchers have also added head tracking to aid their systems [36, 92, 142, 164].

Three dimensional reconstruction has been used by some researchers to aid in building models of human movement during signing [142, 145]. Xu et al used stereo vision methods to detect a set of six non-manual facial markers in Japanese Sign Language with a hit rate of 78.5% and a false negative rate of 10% [152] Video-based motion capture systems were used by Goldstein to track movement of the face during non-manual gestures [50].

A variety of computer vision features are used in ASLR. Ong lists the following categories in his survey paper [105]: two dimensional segmentation, two dimensional moment-based, motion vectors, three dimensional hand positions and three dimensional hand orientations. In addition to these common features I would add: two dimensional head tracking [36, 92, 163], three dimensional head orientation [145], and motion templates [22, 157, 158].

#### *2.3.3.2 Gloves*

Data gloves have been used by researchers for sign language recognition research [69, 79, 84, 98, 96, 135]. These data gloves are usually neoprene gloves with a network of sensors that send detailed information about rotation and movement of the hand and fingers. Data gloves provide detailed sensing about hand configuration and, in many cases, hand and arm movement and rotation. However, data gloves do not provide any information about the non-manual features of signing since they do not provide information about facial gestures or body movement.

One kind of data glove used in ASLR is a electromagnetic motion capture glove [38, 39, 40, 147]. Signers wear gloves with an array of magnetic receivers that are used to provide six degrees of freedom measurements [42]. The positions and rotations are measured absolutely, and orientation in space can be determined. One of the advantages to these systems is that they can provide detailed information about hand configuration and movement in real-time. One of the disadvantages is that common magnetic distortions can create noise in the data.

One of the earliest ASLR efforts with gloves was the work of Kadous in 1996, which

achieved rates of 80% for recognition of isolated Auslan (Australian Sign Language) signs in a 95 sign vocabulary. Vogler and Metaxus used motion capture gloves for the recognition of American Sign Language and achieved a 96.1% recognition rate for a vocabulary of 22 signs using parallel HMMs [146]. The research group of Gao et. al. at Harbin University in China achieved a 92% accuracy on a 5113 sign vocabulary using simple recurrent networks and HMMs.

In addition to commercially available data gloves, some researchers have created custom sensor networks. Reboller created the Acceleglove, a custom arm-based sensor network, which has been demonstrated to recognize finger spelling and 300 ASL signs [59, 95].

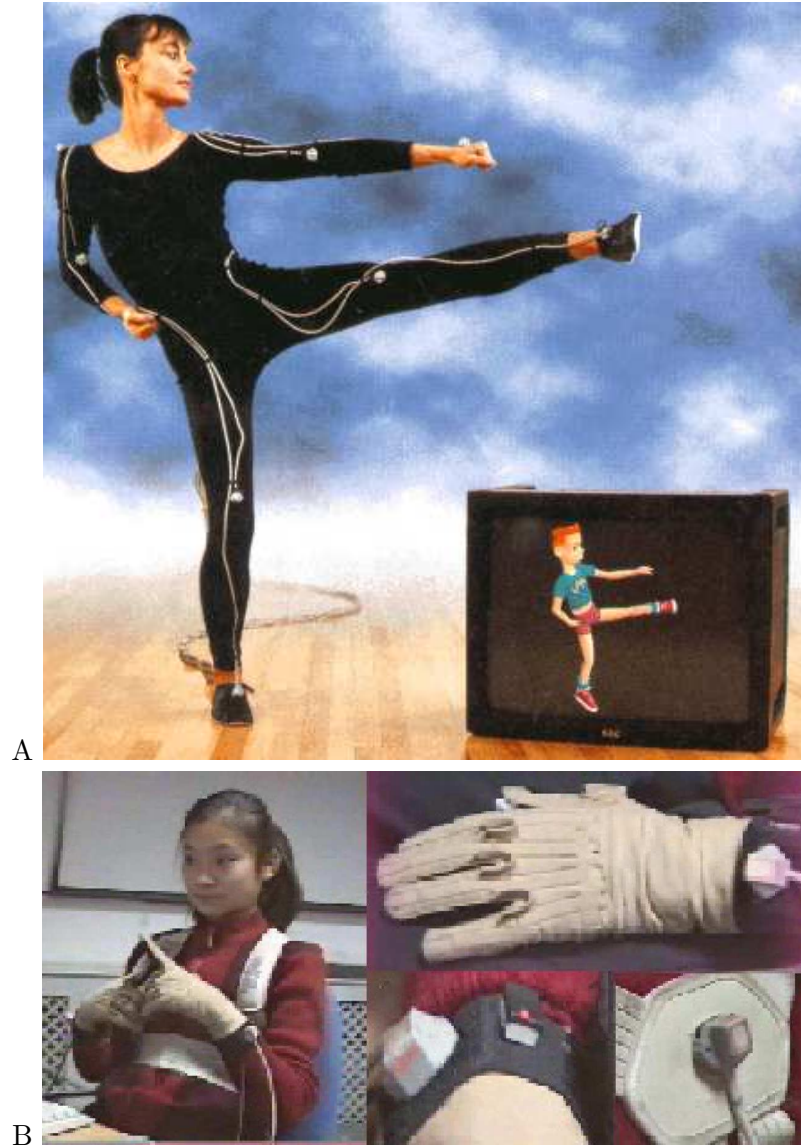
### **2.3.4 Recognition Techniques**

There are a variety of recognition techniques used in ASLR including Hidden Markov models (HMMs), neural networks, rule-based systems, discriminative functions, and hybridized systems. The majority of current research in ASLR uses HMMs [137].

#### *2.3.4.1 HMMs*

HMMs are stochastic models that represent unknown processes as a series of observations. As described in Section 2.2, HMMs have been used in automatic speech recognition with great success. Section 2.2.2 covers their structure and use in automatic speech recognition in greater detail. Gesture recognition researchers have found HMMs to be a useful tool for modeling actions over time [21, 124, 154]. In particular, gesture recognition researchers have had some success with using HMMs for sign language recognition [84, 130, 145]. For an in-depth introduction to HMMs, the interested reader is referred to the tutorial by Rabiner [110].

In the past, researchers have used HMMs to model signing data obtained from various kinds of sensors ranging from single camera systems [130] to data gloves [38] and motion capture systems [142]. Gao et. al. [38] have used data gloves and 3D position trackers to develop a Chinese Sign Language recognition system that achieved a word recognition accuracy of 91.9% on 1500 test sentences with a vocabulary of 5113 signs. It has been shown that ASL recognition can be significantly improved by combining data from two different



**Figure 7:** Polhemus trackers: A) Motion capture system for dance (Image used with permission from Polhemus, Inc.) B) Glove configuration used by Gao et al. (Image used from Fang, Gao, and Zhao “Large-Vocabulary Continuous Sign Language Recognition Based on Transition-Movement Models” with permission [39, 147, 40, 38])

sensors like cameras and accelerometers [17, 95]. Vogler and Metaxus used parallel HMMs to model ASL signs at a phoneme level and achieved an accuracy of 94.3% using a data set of 400 sentences using a vocabulary of 22 signs [143]. Kobayashi and used partially-hidden Markov models to model six Japanese Sign Language signs performed by 20 signers and achieved a recognition rate of 98% [74].

#### *2.3.4.2 Other Classifiers*

The most common alternative to HMMs is the use of neural networks and rule-based systems [105]. Bowden and Cooper have used Markov chains to model signs using small amounts of data [36]. Gao et. al. have used dynamic time warping [44], artificial neural networks (ANN) [45], hierarchical decision trees [40], and simple recurrent networks (SRN) [39]. Matsuo et al. used a rule-based approach to recognize 38 Japanese Sign Language signs performed by two signers and achieved an accuracy of 79% [92]. Hernandez-Rebollar used a decision tree to recognize 30 one-handed signs collected from 17 signers wearing the Acceleglove [59].

### **2.3.5 Basic Unit of Modeling**

Researchers differ greatly in their approach to modeling basic units of signed languages. The simultaneous nature of phonemes in sign languages poses a challenge to many of the sequential techniques that are used in speech recognition [137]. Many researchers choose to use the sign as a base unit of modeling [18, 130], while others attempt to use the structure of phonemes in signing to create models [36, 115].

Vogler and Metaxus have proposed several techniques for handling simultaneous phonemes using the Movement-Hold linguistics model and parallel HMMs [143, 144]. Bowden et. al. use a two-tiered approach which classifies the TAB-SIG-DEZ features from Stokoe’s phonology [132] and passes the results to a Markov chain. Bowden’s two-tiered approach is designed to learn from small amounts of data; he achieved results of 84% on a data set of 49 signs performed in isolation by a single user. When the set was pruned for signs that require non-manual markers or context for meaning, Bowden’s technique achieved an accuracy of 97.67% on 43 signs [36].

Other researchers have focused on techniques that divide signs into sequential components sign as signs or segments [137]. Starner and Pentland demonstrated a system using whole sign modeling that achieved 97.8% accuracy on a 40 word vocabulary using a rule-based grammar [130]. Zieren and Kraiss achieved a 97% accuracy using whole sign HMMs on a vocabulary of 152 German sign language signs performed by a single signer in isolation [165].

#### *2.3.5.1 Sign Transitions*

As discussed in the linguistics section of this chapter, transitions between signs result in a variety of co-articulation effects that are substantially more complicated than speech. Since many of the ASLR projects use data sets composed of isolated signs, few researchers have investigated these transitions. Fang et. al. explicitly modeled these transitions as a series of signs for Chinese Sign Language and achieved a recognition accuracy of 90.8% [38]. Fang's data set contained 51130 sign samples of 5113 isolated signs which were collected from two different signers. Vogler and Metaxus compared several techniques for modeling the movement epenthesis and achieved accuracies of 80%-95% on various data sets [143, 144]

#### *2.3.5.2 Sign Spotting*

One challenge of continuous ASLR in application development is differentiating signs from other background movements. Linguistic research has shown that signed languages gesture and communicative language share the same modality and blur [87]; this shared modality has important implications in the task of spotting signs in videos with other activity as well as differentiating conversational gestures from signs during conversation. Cooper and Bowden used boosted volumetric features (based on Viola and Jones [140]) for sign spotting and achieved recognition rates of around 90% with false recognition rates below 5% on a data set of 5 signs collected from 14 signers [22, 24].

### **2.3.6 Applications of Automatic Sign Language Recognition**

The broader goal of ASLR research is often stated as as improving communication between the Deaf and hearing communities. Many of the research systems seek to combine research

in ASLR, automatic sign language generation, machine translation, and novel interfaces to create sign language based applications for both the Deaf and hearing communities. Below are examples of some ASLR research project applications

- Translation system: Ohki et al. [103] describe a system that used Data glove input for Japanese Sign Language (JSL) recognition and used avatars for signing responses. The system was used for an automatic resident card delivery machine, which has similar interactions to that of an ATM. The system had 11 JSL sentences. The prototype system was evaluated by four deaf users and two interpreters. Users found the system useful but sometimes difficult to understand.
- Information kiosk: Sagawa and Takeuchi [116] describe an interactive JSL kiosk which uses glove based input and avatar animations. The recognition system is based on the group's previous work [61, 115, 118]. The kiosk system is designed to provide information to the public about area businesses and attractions such as restaurants. The system was tested in the Isahaya city office in Nagasaki, Japan for three months. Twenty-seven users participated in the testing, 9 of which were deaf. The study found that 23 users said the system was needed and that 20 of them found the kiosk usable.
- JSL Teaching system: Sagawa and Takeuchi [117] describe a teaching system designed to help non-native signers learn JSL. This system combines signing avatars, Japanese text, iconic representations of gestures (such as a head nod or eyebrow raise) and sign recognition to create an interactive learning experience. This system was evaluated by 24 volunteers with little or no background in JSL. Subjects used the system for 10 minutes and answered questions. Most users had positive feedback on the system. The main issues were difficulty understanding the avatar due to speed and depth perception and that the evaluation of the gestures (with the sign recognition) did not scale well for different body sizes.
- Interactive Dictionary: Athitsos et al. [10] and Dreuw et al. [29] with the Boston University American Sign Language Linguistic Research Project and University of Texas at Arlington are working together to create a system that will allow users

to retrieve sign information using ASLR. The work is a coordinated effort between linguists and computer science researchers.

- **DICTA-SIGN:** The DICTA-SIGN project is an interdisciplinary EU project working on sign language recognition, generation, translation, and modelling [26]. The project includes research on British Sign Language (BSL), German Sign Language (DGS), Greek Sign Language (GSL) and French Sign Language (LSF). The project includes three main applications: sign language-to-sign language terminology translator, search-by-example tool, and a sign language wiki.
- **SignSpeak:** Dreuw et al. [30] describe the goals of the SignSpeak project as the development of a new technologies to translate video sign language examples to text in order to provide more electronic services for the Deaf community and to help improve communication between Deaf and hearing people.

## CHAPTER III

### COPYCAT PROJECT

The CopyCat project was designed to develop an interactive educational adventure game to help deaf children acquire language skills. The main goals of the project are to improve the language and memory abilities of deaf signing children, advance basic research in computer-based sign language recognition, and design an efficient language interaction model in order to assist in the language learning of deaf children. Chapter 2, Section 2.1.3 contains a more detailed introduction to the motivations for the CopyCat project. The CopyCat project was begun as a collaboration between Georgia Tech and the Atlanta Area School for the Deaf in 2004 and has been collecting ASL (American Sign Language) data since Spring of 2005. Since then we have collected 5829 signed phrases from over 30 children. In this chapter I describe the evolution of the CopyCat system design, data collection methodology, and resulting corpus, as well as challenges and successes throughout the process.

#### *3.1 Evolution of CopyCat System*

Our ASL data is collected on site at schools around the Atlanta area. Children play a computer game by wearing colored gloves and signing to characters within the game to accomplish game objectives such as rescuing kittens or defeating villains such as alligators and snakes. Data is collected via wireless accelerometers mounted on the wrists of the gloves and a single video camera. The sensor data is collated and time stamped by the game system and saved as our library for linguistic review and developing our recognition system.

The system has been built in three main phases: game design, data collection, and the ASL recognition engine. Each iteration has been designed with the ultimate goal of moving towards a fully functional system with live recognition that provides productive feedback for students of varying skill levels.

Our corpus collection methods were designed to elicit live, casual signing from children

as they interact with characters in the game. This approach has resulted in a data set that contains many language modeling challenges including disfluencies, pauses, dominant hand switching, and sign variations. Our research has focused on developing labeling schemes and training models to accurately reflect the children’s signing.

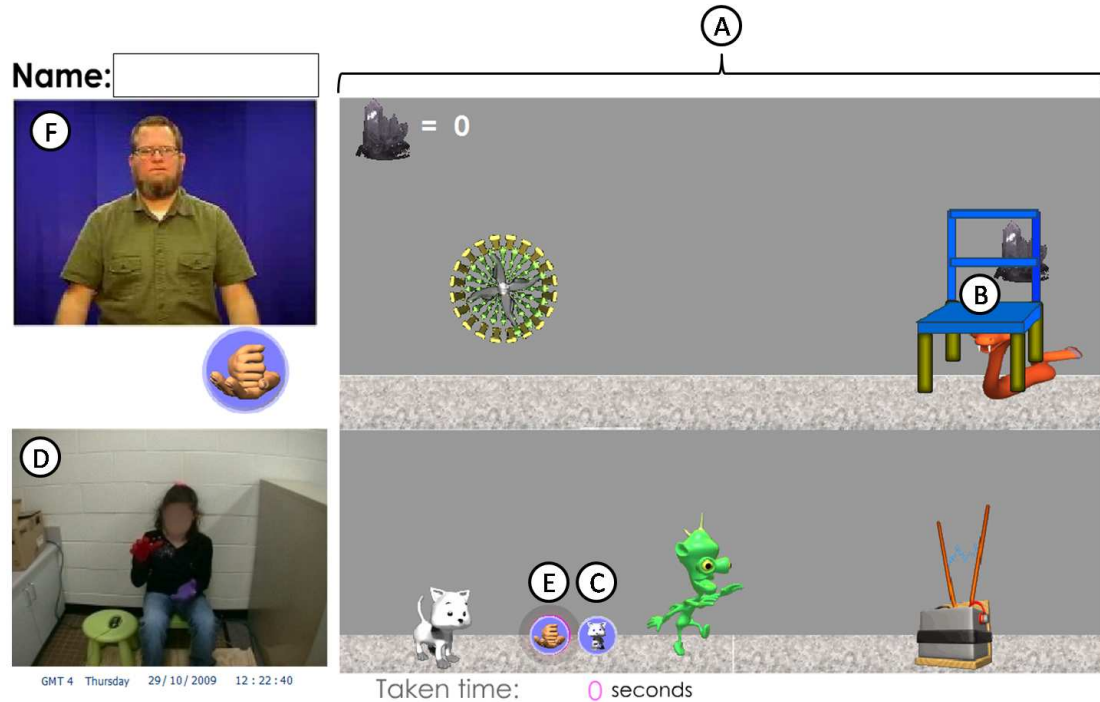
### **3.1.1 The CopyCat Games**

As part of the CopyCat project, several computer-assisted language learning games have been designed. Each game entails some sort of quest by the hero to collect items in order to solve a problem. In each quest, the child interacts with the hero via sign language to warn them of a villain or identify where a hidden object is located. If the children know what to tell the hero regarding the guards location they can use the mouse to click a “talk” button to turn the hero towards them so they can sign to her. They then click the “talk” button again when they are finished signing. If the child is uncertain what to say they can click a “help” button to see the tutor in the top left corner of the screen tell them what to say. The child may view the tutor repeatedly if they so choose (see Figure 8).

After the child talks to the hero, the child’s signing is classified as correct or incorrect. If the child’s signing is incorrect, a question mark appears above the hero’s head to simulate misunderstanding by the hero, and the child must try again to communicate accurately. If the child’s sign is correct, the hero, with the wave of a paw, “poofs” the guard, turning it into an innocuous item, and the hero continues on the quest.

### **3.1.2 Language Learning**

The video tutor examples in the game were designed to be similar to a communication setting which young children encounter while learning language through interaction with adults. As the child’s linguistic and communicative competence and confidence grow, the need for such assistance diminishes and the child can respond appropriately without help. Thus, our tutor performs the role of the good adult language model [123], always available to the child and responding to the child’s cue (in this case a press of the “help” button) in an appropriate linguistic manner.

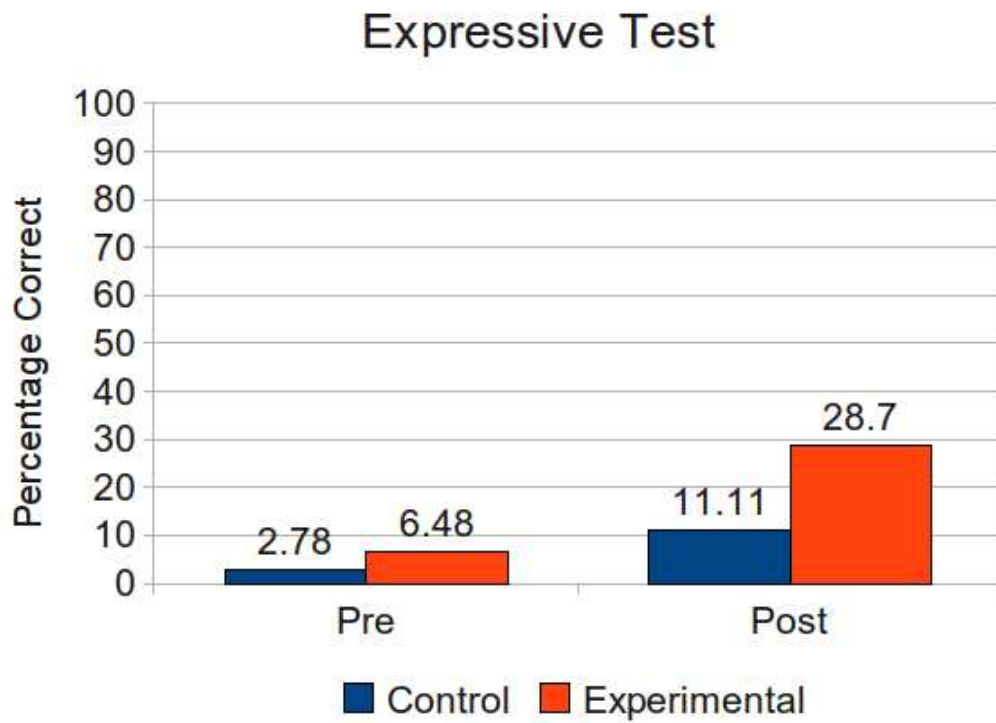


**Figure 8:** CopyCat screen shot from Mini Quests: A) Animated game characters in their worlds B) The villain (a snake) is hiding behind the chair. C) The push-to-sign button has a picture of the hero (Iris the cat) on it. Children push the button to sign to the hero and warn her about the snake. D) The live video feed allows children to see themselves as they sign. E) The help button has a picture of the sign for help and will cue ASL video to help the children during game play. F) This window displays the tutor videos.

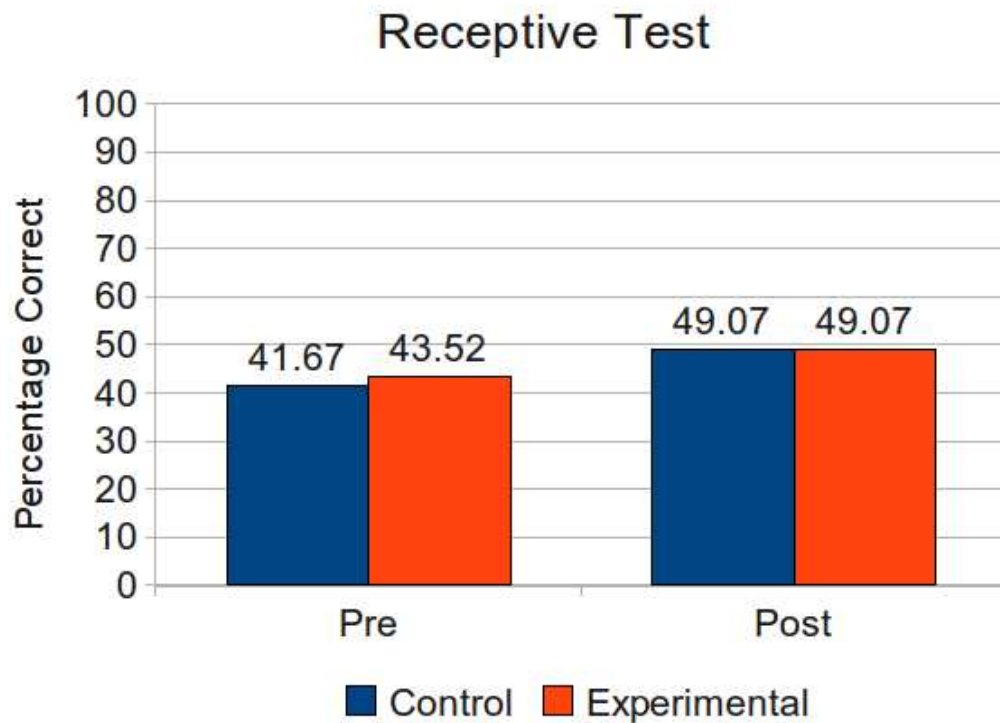
### 3.1.3 Educational Evaluation

In order to collect data regarding the language processing abilities of the children and the efficacy of the game’s language interaction model, pretests and post tests were administered and in-game response data were recorded. These tests consisted of sections to test expressive language skills (results shown in Figure 9), receptive language skills (results shown in Figure 10), and working memory (results shown in Figure 11).

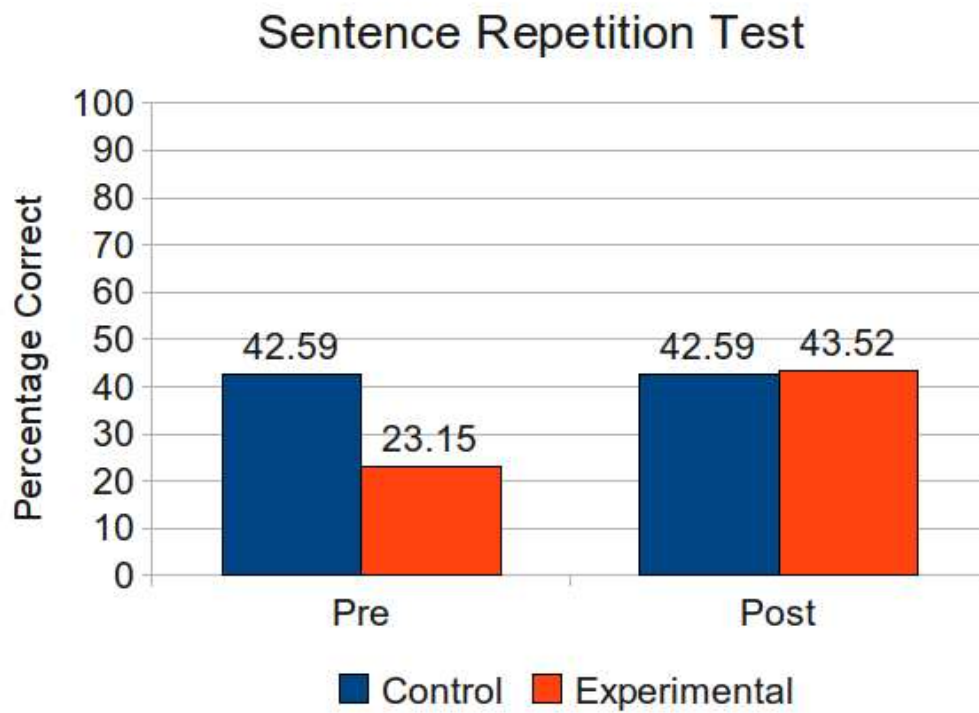
The results of the expressive language test indicate that the experimental group made a significant gain in the accuracy of their signing when describing a video they saw as well as in their length of utterance as measured by mean length of utterance from pretest to post test [149].



**Figure 9:** Results of expressive language tests during educational evaluations for the CopyCat Phase 2 deployments



**Figure 10:** Results of the receptive language tests during educational evaluations for the CopyCat Phase 2 deployments



**Figure 11:** Results of sentence repetition tests during educational evaluations for the CopyCat Phase 2 deployments

## ***3.2 System Design***

### **3.2.1 Interface Design**

The iterative design cycle allows us to adapt to problems as they emerge during the development process and has allowed the CopyCat system to improve rapidly. Our user interface for game play uses a video stream of the user, feedback from characters in the game, and help videos in ASL to engage the children. The live video stream allows the children to see their signing and engages them in the signing. The children enjoy “being in the game” and tend to use the feedback to stay in frame.

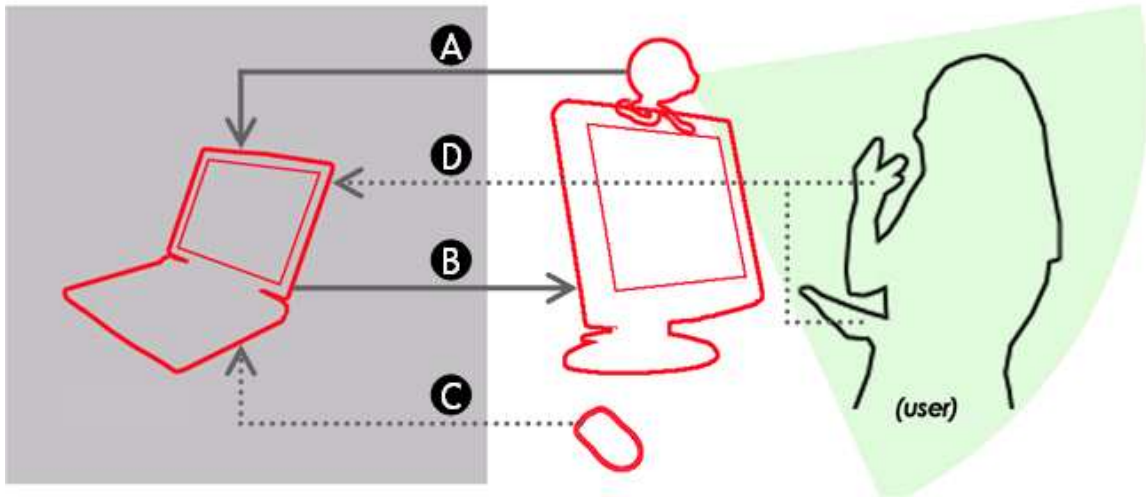
The game characters have been designed to attempt to elicit natural signing. When the child clicks the “talk” button, the character will face the child and pay attention while they are signing. If the signing is incorrect, a question mark thought bubble shows above the character. We have found that visual clues such as these help guide the children in their interactions.

The introduction instruction and game help videos are all ASL. We have taken care to synchronize the spatial layout of the game with the spatial constructs in signing to provide consistency. Even simple modifications to the interface, such as moving a button require a check of all of the ASL spatial referencing in the videos.

### **3.2.2 Wizard of Oz**

When the functionality of a system is under development, developers can sometimes replace that functionality with a person, similar to the “Great Wizard of Oz” operating behind the curtain. The system can be tested while the hidden “wizard” controls operations and developers can obtain critical feedback about system design early in the process [28].

We divided game development and sign language recognition by using a “Wizard of Oz” setup, shown in Figure 12 [58]. The child interacts with the user computer (on the right) by navigating with the mouse and signing to characters. The wizard’s computer (on the left) controls the game’s response to children signing and collects data from the sensors and game logs for future use.



**Figure 12:** Diagram of the Wizard of Oz system system showing A) live camera and sensor feed B) interface output split between wizard and user C) child’s mouse and D) the interface computer

### 3.2.3 Sensors

The CopyCat system uses computer vision and three-axis accelerometers to collect data for use in sign language recognition. Our computer vision is processed from video collected on a single IEEE 1394 DV camcorder that faces the children. The children wear colored gloves, which contain small accelerometers mounted on the outside of the wrist (shown in Figure 13). These accelerometers provide information on movement: acceleration, direction, and rotation of the hand. The distinct color of the gloves helps distinguish the hands from the skin color of the face and cluttered backgrounds. The wizard’s computer coordinates the data streams, synchronizes them, and stores them for future use.

One key design goal has been to have a portable system that will work in a variety of environments. Our deployment environments at the schools have ranged from classrooms and libraries, to a re-purposed supply closet. Figure 14 shows the construction of a “signing kiosk” and the resulting view from the camera. The kiosk is inexpensive and modular so that it can be transported easily. The kiosk fixes the position of the camera relative to the child’s position on the chair. The color of the furniture can be used to help calibrate the video camera’s color balance to enable more accurate hand tracking. This kiosk design



**Figure 13:** Gloves with accelerometer (top). Detail of wrist-mounted accelerometers (bottom).

allows us to move our equipment from area to area with minimal re-calibration.



**Figure 14:** Kiosk

### ***3.3 Resulting Corpus***

#### **3.3.1 Overview of Data Collected**

Each phase of the CopyCat project includes on-site deployments to collect data at our two partner schools: Atlanta Area School for the Deaf and Gwinett Independent School District. We have collected a total of 5829 phrases over four phases, with a total of nine deployments. Table 1 shows a count of phrases collected throughout the CopyCat project. Each phrase is a three, four, or five sign sentence taken from a vocabulary of 22 signs. The phrases are of the format *[adjective1] subject preposition [adjective2] object* .

#### **3.3.2 Characterizing the Children’s Signing**

Most of the sign language databases used for automatic sign recognition are carefully scripted and collected in a controlled environment [137]. Our data set provides many samples of children signing as they interact with the online characters. This signing contains many of the artifacts of conversational signing such as disfluencies like pauses, false starts, hesitations, and sign variations. It also has many examples of non-signing activities such

**Table 1:** Table of data collected during the CopyCat project

<b>Phase</b>	<b>Game</b>	<b>Date</b>	<b>Users</b>	<b>Ages</b>	<b>Total Phrases</b>
Pilot	Kitten Escape!	Spring 2005	3	9-11	50
Pilot	Kitten Escape!	Spring 2005	2	9-11	78
<b>Total</b>					<b>128</b>
First Deployment	Kitten Escape!	Spring 2005	5	9-11	627
First Deployment	Castle Quest	Fall 2005	9	9-11	1812
<b>Total</b>					<b>2439</b>
Second Deployment	Mini Quests	Fall 2008	5	6-11	505
Second Deployment	Mini Quests	Spring 2009	5	6-11	503
Second Deployment	Mini Quests	Spring 2009	14	6-11	822
<b>Total</b>					<b>1830</b>
Third Deployment	Mini Quests	Fall 2009	11	6-9	1432
<b>Total</b>					<b>1432</b>
<b>CopyCat Total</b>					<b>5829</b>

**Table 2:** Game vocabulary

<b>Subject</b>	<b>Object</b>	<b>Adjective</b>	<b>Verb</b>
ALLIGATOR	BED	BLACK	BEHIND
CAT	BOX	BLUE	IN
SNAKE	CHAIR	GREEN	ON
SPIDER	FLOWERS	ORANGE	UNDER
	WAGON	WHITE	
	WALL		

as scratching and fidgeting.

The conversational nature of the children’s interaction with the game’s characters results in signing samples that contain signing beyond basic game vocabulary. The data set contains many non-game communications towards game characters (including messages such as WRONG or RED NO BLUE), signs that are not in the game vocabulary, and even gestures that are not ASL such as a wave which is used generally to indicate an error and restart (a kind of “erase” gesture).

The children’s signing handedness did not directly correspond to their dominant handedness for other activities and was inconsistent even within the phrases. This hand switching makes it more difficult to group signs and phrases by handedness for modeling purposes. Dominant hand switching is probably a symptom of their low fluency and is common among children [90].

### ***3.4 Challenges of the CopyCat Corpus***

#### **3.4.1 Library Continuity**

There is a continued tension between goals for system improvement, expansion of game functionality, and library expansion. Though our upgrades in sensors and configuration have improved the reliability and portability of the system, they also detract from backwards compatibility (from the perspective of automatic sign language recognition). This discontinuity results in a larger corpora of children signing, with sub-sets from various deployments that are incompatible with each other.

The library data is stored in both its raw format as well as a format that includes post-processing from vision and accelerometer sub-routines. This redundancy in storage requires more disk space, but helps alleviate the continuity problems by allowing for changes in post-processing without losing entire library sets. For example, we have changed our computer vision code several times. The raw data library allows us to experiment different with post-processing schemes and choose optimally.

### **3.4.2 Sensor Changes**

During the design cycle we have changed the sensors several times. Two of our main design priorities are system reliability and system portability. Our long term goal is a system that can be deployed at any school and requires minimal maintenance. We started with the explicit goal that our sensors be inexpensive and easy for schools to use.

During the project, we have used both commercially available accelerometers and those we designed in-house. We have iterated on accelerometer collection code in order to address issues that emerged with calibration, output normalization, and sensor drift [150]. These changes, combined with changes to the video frame rate, create incompatibilities with existing data from previous deployments. The cycle of updating technology is a further challenge to library continuity, and such changes must be carefully considered. In this work I use the data set collected from the second phase deployment, which has consistent hardware and configuration during the entire deployment.

### **3.4.3 Varied Environments**

Each time we visit a school for a deployment, we have no guarantees where they will have space for the system. These changes in environment create challenges for computer vision algorithms. Many sign language recognition systems depend on very static environments for their algorithms to work. We have worked to make the system more portable by a combination of choosing more flexible algorithms and creating an environment where visual cues can help keep the algorithms calibrated. The kiosk helps ensure that the camera distances are approximately the same each time. Additionally, the colored gloves and furniture help provide reference points for algorithms to track hands, face, and body movement in the video frame.

### **3.4.4 Data Integrity**

During data collection the system must coordinate data streams from three different sensors. These streams must be saved to disk, logged, and synchronized. One of the challenges of this configuration is keeping the data streams synchronized and providing live feedback for errors

in reading, synchronizing, or logging sensor data. Our most recent iteration has focused on creating a subsystem specifically to provide feedback to administrators to prevent problems in game play and data loss.

Post-processing of the libraries can also discover errors in the data stream. These errors must be diagnosed for future prevention, and the samples must be catalogued as damaged data.

#### **3.4.5 Automatic Annotation**

We have designed the game to provide as much automatic annotation as possible to help us index and use our data. Each signed phrase contains logs with information on user, session details, wizard feedback, and game information. After the data is stored, our post-processing is also largely automated. These logs provide further information about the content of the signed phrase. All of this data helps us rapidly compile statistics on the data set and select sub-sets by interesting features.

#### **3.4.6 Maintaining the Library**

As the library increases in size and complexity, we have continued to try to address issues with maintaining our data. Maintaining logs and raw data allow us to continue to do retrospective evaluations of many aspects of the process. There are different research and publication cycles for the various topics of the CopyCat project: computer vision, machine learning, human-computer interaction, sign linguistics, and education.

The size of the data has been growing since the beginning of the project. Not only does each deployment add more data instances to the library, but the size of the data per instance has been growing as well. Verifying the integrity of automated process logs is tedious and is time consuming. We have increased the sensor sampling rate, as well as the detail and complexity of the game logs. Additionally, we must keep track of data from educational testing which includes a large amount of video of the children's language testing sessions.

### **3.4.7 Sign Variation**

The machine learning system needs many examples of the same signs across many users for building representative models that are robust to variations. Thus far we have maintained a fairly small vocabulary, which allows for many examples of each sign. Even with the small vocabulary, we have discovered that there are often many variations on how a sign is performed. Most of these variations are technically correct, and we should make allowance for them. If only one or two children perform a specific variation, it can make collecting sufficient examples difficult.

### **3.4.8 Influencing the Children’s Signing**

Throughout the iterations of game design, we have continued to create an interface that influences how the children sign. The story line of the game helps restrict vocabulary by limiting the scope of objects and characters on the screen for the children to describe or address. By creating a conversational environment, we can influence how the children sign.

The push-to-sign approach to the game has a dual purpose of segmenting the signing sequences and giving the children pause to focus. We have even found that children will sometimes take a moment to rehearse their signing before clicking to get the character’s attention in the game. These techniques have greatly improved the quality of signing we get from the children, but we still face challenges with out-of-vocabulary signs and the children’s difficulties in performing the signs correctly.

### **3.4.9 Privacy Issues**

Because our data is collected from children, our data is subject to strict privacy requirements. Our long-term goal is to make sections of the data available to linguistic and machine learning researchers. Anonymizing the video data compromises the content, since the face is the center of the signing space and facial gestures are a component in ASL. We have been working with our institutional review board and the host schools to create an agreement that would allow us a mechanism to release data to other researchers.

### ***3.5 Summary***

CopyCat is a long-term project that has used an iterative development to design an interactive, educational game for deaf children. Designing and deploying the game for user testing created unique challenges in collecting, storing, and using the large data set of children's signs. We have addressed many of these challenges with strategic game improvements generated from the feedback phase of the iterative cycle.

As CopyCat matures into a commercial-grade system, we are focusing on long-term library collection and management. The success of CopyCat will depend on our ability to easily integrate new data from each deployment into our library. We are focusing on ways to automate the collection and indexing of data for storage in a central library. As we build models off of the central library, each deployment site will get updates to the game recognition system.

## CHAPTER IV

### LABELING

In this chapter I will describe the development of my labeling ontology, the process of labeling the data, and an analysis of the labeled data. The labeling ontology was developed with the intent of developing a greater understanding of the contents of our data set and the goal of improving the annotation schemes for modeling the children's signing for automatic sign recognition. The development of the ontology was an iterative effort that integrated feedback from the automatic recognition, linguistics and game play perspectives; the process reflects the interdisciplinary nature of the work.

The role of conversational repair is of particular interest when considering artifacts of the children's signing in our data set since many of the variations in signing are a result of conversational repair. We can classify the conversational repairs by the children as both self-repairs and other-repairs [125]. The children's self-repairs are evident in the self-corrections that occur during signing such as the use of the erase wave (discussed further in Section 4.3.2) and the repetition and restarting of signing. The game feedback of correct or incorrect can be seen as a mechanism of other-repair, since a classification of incorrect results is a failure to accomplish the game task. This feedback conveys a lack of understanding and requires the student to initiate the signing task again.

The two most common kinds of conversational repair that occur during the game play were replacement repairs and word-search repairs. Replacement repairs occur when the child replaces erroneous signing either through a phrase restart, the addition of a correct sign to replace an incorrect sign, or, in the case of the other-repair function of the game, re-try of failed attempts. Word-search repairs are frequently seen as a "laundry list" effect, where students list several colors in sign before either deciding which color is the correct one or hoping that in listing all colors they provided one which was correct. Other word-search repairs can be seen as students sign to themselves as they search for the correct sign or

phrase. For a broader discussion of conversational self-repair see Chapter 2, Section 2.1.9.

## **4.1 Research Goals**

The work described in this chapter can be summarised as

- Contributions:
  - I will identify significant gestures in our dataset, including game vocabulary, relevant non-game signs, communications directed towards game characters, and disfluencies in sign. I will enumerate and model these significant gestures.
  - I will use this ontology to characterize the data set.
- Research Questions:
  - What are the signs, behaviors, and variations that occur in the data set?
  - How should these artifacts be labeled?
  - Can we use the labels to characterize our data set?
  - What are the frequency and distribution of signs and gestures within the data set?
- Hypotheses:
  - A review of the data set will show variations in dominant hand usage, variations in sign construction, non-game vocabulary, and non-sign activities.
  - The signs and gestures present in the children’s signing can be characterized by sign label, handedness, and quality
  - The labeled transcripts of the children’s signing will provide important information about the children’s usage of ASL signs and grammar, as well as the non-sign content of their signing.
- Methods:
  - I conducted a manual review of the data set, first by taking notes on sign videos and finally by labeling subsets of the data. These labeled subsets will be used to refine the labeling ontology iteratively. The ontology was defined as a collaboration with our linguist.

- I examined the samples for labeling and compiled a list of criteria for each label.
- I used the ontology to manually inspect and label each sample in the data set.
- Data Collected:
  - CopyCat data from the second deployment: Fall 2008, Spring I 2009, and Spring II 2009
  - Set of labels and descriptions for sign label, handedness, and quality.
  - Hand-labeled transcripts for each sample that include time segmented labels for sign, quality, and handedness
- Analysis:
  - Comparison of variations in the children’s signing to the game vocabulary and grammar
  - Inspection of the video and comparison of labels
  - Aggregate statistics about the labels which provide information on frequency and distribution of artifacts throughout the data set

## ***4.2 Data Set***

A summary of the full CopyCat corpus can be found in Table 1 from Chapter 3. Data from the second deployment was used for this dissertation work. The second deployment data was collected over three different sessions: Fall 2008, Spring 2009 I, and Spring 2010 II. Tables 3, 4, and 5 show the signed phrases for the “Mini Quests” game which was used for the second deployment.

**Table 3:** Level 1 phrases used in the CopyCat game data. Signs in brackets are optional signs.

<b>Encounter</b>	<b>Phrase</b>
0	[GREEN] ALLIGATOR BEHIND [BLUE] WALL
1	[ORANGE] SNAKE BEHIND [BLUE] WALL
2	[ORANGE] SPIDER ON [BLUE] WALL
3	[ORANGE] CAT ON [BLUE] WALL
4	[ORANGE] SNAKE UNDER [BLUE] CHAIR
5	[ORANGE] SPIDER ON [BLUE] CHAIR
6	[GREEN] ALLIGATOR BEHIND [BLUE] CHAIR
7	[ORANGE] CAT UNDER [BLUE] CHAIR
8	[GREEN] ALLIGATOR IN [BLUE] BOX
9	[ORANGE] SPIDER IN [BLUE] BOX
10	[ORANGE] CAT BEHIND [BLUE] BOX
11	[ORANGE] SNAKE ON [BLUE] BOX
12	[ORANGE] CAT BEHIND [BLUE] BED
13	[GREEN] ALLIGATOR ON [BLUE] BED
14	[ORANGE] SNAKE UNDER [BLUE] BED
15	[ORANGE] SPIDER UNDER [BLUE] BED
16	[GREEN] ALLIGATOR IN [BLUE] WAGON
17	[ORANGE] SPIDER UNDER [BLUE] WAGON
18	[ORANGE] CAT BEHIND [WHITE—ORANGE] FLOWERS
19	[ORANGE] SNAKE IN [WHITE—ORANGE] FLOWERS

**Table 4:** Level 2 phrases used in the CopyCat game data. Signs in brackets are optional signs.

<b>Encounter</b>	<b>Phrase</b>
20	[GREEN] ALLIGATOR ON BLUE WALL
21	[ORANGE] SPIDER IN GREEN BOX
22	[BLUE] SPIDER IN ORANGE FLOWERS
23	[ORANGE] SNAKE UNDER BLUE CHAIR
24	[GREEN] ALLIGATOR BEHIND BLUE WAGON
25	[ORANGE] SNAKE UNDER BLACK CHAIR
26	[ORANGE] CAT ON BLUE BED
27	[ORANGE] CAT ON GREEN WALL
28	[GREEN] ALLIGATOR UNDER GREEN BED
29	[ORANGE] SPIDER ON WHITE WALL
30	[ORANGE] SPIDER UNDER BLUE CHAIR
31	[GREEN] ALLIGATOR IN ORANGE FLOWERS
32	[ORANGE] CAT BEHIND ORANGE BED
33	[GREEN] ALLIGATOR BEHIND BLACK WALL
34	[ORANGE] SNAKE UNDER BLUE FLOWERS
35	[ORANGE] CAT UNDER ORANGE CHAIR
35	[ORANGE] SNAKE IN GREEN WAGON
36	[ORANGE] SNAKE IN GREEN WAGON
37	[ORANGE] SPIDER IN BLUE BOX
38	[GREEN] ALLIGATOR BEHIND ORANGE WAGON
39	[ORANGE] CAT UNDER BLUE BED

**Table 5:** Level 3 phrases used in the CopyCat game data

<b>Encounter</b>	<b>Phrase</b>
40	BLUE ALLIGATOR ON GREEN WALL
41	ORANGE SPIDER IN GREEN BOX
42	BLACK SNAKE UNDER BLUE CHAIR
43	BLACK ALLIGATOR BEHIND ORANGE WAGON
44	GREEN SNAKE UNDER BLUE CHAIR
45	BLACK SPIDER IN WHITE FLOWERS
46	BLACK CAT ON GREEN BED
47	WHITE CAT ON ORANGE WALL
48	GREEN ALLIGATOR UNDER BLUE FLOWERS
49	BLUE SPIDER ON GREEN BOX
50	GREEN SNAKE UNDER BLUE CHAIR
51	ORANGE ALLIGATOR IN GREEN FLOWERS
52	BLACK CAT BEHIND GREEN BED
53	WHITE ALLIGATOR ON BLUE WALL
54	ORANGE SNAKE UNDER BLUE FLOWERS
55	GREEN SPIDER UNDER ORANGE CHAIR
56	BLACK CAT IN BLUE WAGON
57	WHITE CAT IN GREEN BOX
58	WHITE SNAKE IN BLUE FLOWERS
59	ORANGE SPIDER UNDER GREEN FLOWERS

### 4.3 *Criteria*

#### 4.3.1 **Developing the Criteria**

The criteria for defining the ontology began from simple sign-based labeling and evolved iteratively. I designed the criteria by beginning with the basic scheme and reviewing the data. During each review I made notes on video content and discussed the criteria with our linguist and other members of the research group. At each iteration, the notes and feedback were integrated into the criteria definitions until the criteria was consistent with the data set. Signs were labeled for the following dimensions:

- **Sign label** to indicate the ASL sign.
- **Handedness** to indicate the dominant hand for the sign
- **Quality** to indicate the accuracy of the sign.

#### 4.3.1.1 *Quality*

One of the more difficult questions was defining the quality of signing and what constituted a correct example of the sign. There is a broad spectrum of signing quality within the data set. This continuum had to be divided into GOOD, OK, BAD, and not acceptable. Signs that are not acceptable are put into the “garbage” class for partially articulated signs and gestures that are not understandable. The dividing line between BAD and not acceptable was based heavily on what the linguist had accepted during game play. There are a number of nonsensical signs present in the data set that had major construction errors and were unintelligible.

The dividing line between GOOD and OK was fairly easily derived as roughly the division between precisely performed signs and signs that are easily understandable but allow for the kinds of variation commonly seen in conversational signing. The children were putting significant effort into signing well to the computer, so we have a large number of signs that were classified as GOOD.

#### 4.3.1.2 *Non-signs*

Within the non-sign gestures several sub-groups emerged: clearly defined sign-like gestures, silences, and non-sign activities. The clearly defined sign-like gestures were divided into two mouse gestures (start of the sentence and end of the sentence), the erase wave, and garbage signs. The erase wave was surprisingly consistent for the children, with its major variation being its handedness.

The mouse gestures had been loosely defined previously as markers for starting and ending sentences in previous work [18], but in this work were more clearly defined as the gestures moving from the mouse to signing space and moving from the signing space to the mouse. These definitions were needed because several children moved the mouse around during the data collection, which resulted in a `START_SENTENCE` and `END_SENTENCE` gesture in the middle of a phrase. These extra mouse gestures were sometimes before a self-correction or used as a delay to think about their response.

The garbage class was needed to encompass the activities that looked like signing, but

were nonsense. These signs most often occurred as an attempt to “fake it” when children were uncertain, either as a fake sign that is rushed through, or an entry on a list of signs such as “BLUE GREEN GARBAGE ORANGE YELLOW.” The listing of signs was done frequently when the children were trying to decide which was the correct answer and in many cases could be thought of as them talking to themselves. The garbage class also included some non-sign activities that occurred inside the signing space such as sneezing or coughing.

A fidget class was needed to differentiate activities that were not in the signing space and were full body activities. The children often shifted in their chair, stretched, and wiggled during the sessions. These whole-body movements were separated into their own class for the benefit of more accurate models.

#### *4.3.1.3 Out of Vocabulary Signs*

Almost all of the out of vocabulary signs were the Signed Exact English (SEE) signs for THE and IS. Though the schools where the data was collected have ASL-based curricula, the children are exposed to SEE at various places including previous schools. Additionally, the children occasionally used a SEE sign for chair, which uses an initialized C-hand instead of an H-hand [41, 53]. The ASL sign WRONG was used by several students as part of their self-corrections.

### **4.3.2 Criteria: Sign Label**

The sign label criteria began with a basic gesture based approach of one sign label per gesture and was expanded from there. The children have varied language backgrounds, which resulted in a number of variations of signing production, as well as some out-of-vocabulary signs. Out-of-vocabulary signs were added to the sign labels, and a breakdown of these are shown in Table 6.

There were also several variations of the signs in the game vocabulary, such as the word chair. ASL uses a “H” hand to sign CHAIR and SEE uses a “C” hand. Figure 15 shows examples of both signs. Both variations are acceptable for CHAIR, but are distinctly different signs. Start sentence and end sentence were needed to capture information about



**Figure 15:** Snap shots of signs for CHAIR from two different children. The left example shows the ASL sign for chair, which uses two H-hands. The right example shows the SEE sign for chair, which uses an H-hand for the dominant (active) hand and a C-hand for the non-dominant (passive) hand.

the students’ use of the mouse within the game.

The following summarize the non-sign labels I used in the ontology:

- **SILENCE:** Any lack of movement between signs that lasts longer than 5 frames is labeled as SILENCE. Smaller pauses between signs are considered part of the co-articulation and absorbed by the surrounding signs.
- **GARBAGE:** Any fake signs are labeled as GARBAGE. This label most commonly includes segments where the children are trying to remember a sign and cycle through various hand shapes and movements while they are thinking “out loud.” Also included are signs where the children do not know the sign and move randomly while trying to fake the sign.
- **FIDGET:** Any whole-body movement that occurs between signs. This class predominantly consists of children repositioning themselves on the chair, leaning forward to look at the screen, and leaning back to get comfortable.
- **WAVE:** This is a wave which usually indicates an erase gesture. The gesture is performed both one- or two-handed, with an open 5-hand back and forth in front of the signer. The 5-hand can be either facing towards or away from the body. Sometimes



**Figure 16:** Some examples of non-sign gestures. From left to right: sneeze, head scratch, chin pause, and WAVE.



**Figure 17:** START\_SENTENCE (left) begins with the hand on the mouse, and then the hand moves into the signing space. END\_SENTENCE (right) begins with the hand in the signing space, and then the hand moves to the mouse.

children wave as a filler while thinking, similar to saying “um.” Examples of the erase wave sequence are shown in Figure 18.

- START\_SENTENCE: This label is used for labeling the gesture where students remove their hand from the mouse and move it into the signing space.
- END\_SENTENCE: This label is used for labeling the gesture where students move their hand from the signing space to the mouse and click the mouse to end signing.

#### 4.3.3 Criteria: Handedness

In order to discuss the handedness of the children’s signs, it is important to understand the relevant terms. In Sandler’s discussion of handedness of sign, she defines the dominant hand as the primary articulator. The non-dominant hand’s actions during signing are limited to



**Figure 18:** WAVE gesture from four different children. A) one-handed / right hand / hand facing away from body B) one-handed / left hand / hand facing away from body C) two-handed / hands facing away from body D) two-handed / hand facing towards body,

**Table 6:** A comparison of the original game vocabulary and labels that were added.

Labels	Game	Added
Subject	ALLIGATOR, CAT, SNAKE, SPIDER	
Object	BED, BOX, CHAIR, FLOWERS, WAGON, WALL	
Article		THE
Adjective	BLACK, BLUE, GREEN, ORANGE, WHITE	CHAIR_CHAND
Verb	BEHIND, IN, ON, UNDER	IS
Non-sign		SILENCE, GARBAGE, START- SENTENCE, END-SENTENCE, ERASE-WAVE, FIDGET, WRONG

non-action, copying the dominant hand or acting as a place of articulation [9, 119]. Battison states these rules as the symmetry and dominance conditions [12]:

- **Symmetry condition:** If both hands of a sign move independently during its articulation, then both hands must be specified for the same location, the same hand shape, and the same movement (whether performed simultaneously or in alternation), and the specification for orientation must be either symmetrical or identical.
- **Dominance condition:** If the hands of a two-handed sign do not have the same hand shape (i.e., they are different), then one hand must be passive while the active hand articulates the movement; the specification of the passive hand is restricted to a small set: A-hand, S-hand, B-hand, 5-hand, G-hand, C-hand, and O-hand.

When the non-dominant hand acts as a place of articulation, it may assume its own hand shape and location, which the dominant hand uses as a reference point. For the sign CHAIR, the non-dominant hand uses an H-hand located in front of the torso, with palm orientation down. The dominant also uses an H-hand with the palm facing down and taps the index finger tips together. In this manner, the non-dominant hand acts as a passive location for the dominant hand to touch.

The following definitions help put these concepts together in instruction for signing and were taken from the popular “Green Book” for teaching ASL[11]:

- **Dominant hand:** In a right-handed signer, the right hand is the *dominant hand*.
- **Active hand:** The hand that moves when making a sign (as opposed to the *passive* hand). For example with a right-handed signer, the right hand is *active*, when making the sign MONEY. However, both hands are *active* in the sign EXCITED.
- **Non-dominant hand:** In a right-handed signer, the left hand is the *non-dominant hand*.
- **Passive hand:** The hand that does not move when making a sign (as opposed to the *active hand*). For example for a right-handed signer, the left hand is passive when making the sign MONEY. The *passive* hand is also sometimes called the *base hand*.

The signs were labeled individually by the dominant and active hands. One-handed signs are labeled solely by the dominant hand. Two-handed signs that have an active and a passive hand are assigned handedness based on the active hand. Some two handed signs are symmetric and do not have a dominant hand, since both hands are active and mirror each other.

The following definitions were used for the handedness labeling:

- **RIGHT:** sign which is performed with only the right hand
- **LEFT:** sign which is performed with only the left hand
- **BOTH\_RIGHT:** sign which is performed two-handed with the right hand as the active hand
- **BOTH\_LEFT:** sign which is performed two-handed with the left hand as the active hand
- **BOTH:** sign which is performed two-handed and is a symmetric sign (has no dominant hand)

#### 4.3.4 Criteria: Quality

The quality of signs was rated in order to provide a better understanding of the variety of signing present in the data set. These ratings were used to qualify information about the fluency of the children's signing. Additionally, the quality ratings were also used to divide the data into pools for testing and training.



**Figure 19:** CAT from three different children. The left examples show CAT performed right handed. The center example shows CAT performed with both hands (BOTH symmetric). The right example shows CAT performed left handed.

The following definitions were used for the quality labeling:

- **GOOD:** These signs are easily recognizable at normal speed play. They obey the structure of the intended sign and are basically “textbook” signing.
- **OK:** These signs are recognizable as the sign at normal speed play, but may have poor form or variation in the sign. Signs which have internal pauses or unusual fidgeting are generally in this class.
- **BAD:** These signs are barely recognizable as the intended sign and may require multiple views or slow motion to determine the intended sign.

**Table 7:** Breakdown of sign distribution by variations in handedness. Variations in some signs result in their placement in multiple categories.

Labels	One hand RIGHT, LEFT	Two hand: Asymmetric BOTH-RIGHT, BOTH-LEFT	Two hand: Symmetric BOTH	Non-handed
ALLIGATOR		✓		
BED	✓	✓		
BEHIND		✓		
BLACK	✓			
BLUE	✓			
BOX		✓		
CAT	✓		✓	
CHAIR		✓		
CHAIR-CHAND		✓		
END-SENTENCE				✓
WAVE	✓		✓	
FIDGET				✓
FLOWERS	✓			
GARBAGE				✓
GREEN	✓			
IN		✓		
IS	✓			
ON		✓		
ORANGE	✓			
SILENCE				✓
SNAKE	✓			
SPIDER		✓		
START-SENTENCE				✓
THE	✓			
UNDER		✓		
WAGON		✓		
WALL			✓	
WHITE	✓			
WRONG	✓			

## 4.4 *Labeling Tool*

### 4.4.1 Design

I designed “Data Labeler” and created, a tool to aid in the labeling of large set of data of sign data for recognition purposes. The design goals of the tool were

- **Video editing and replay:** The video for each sample is used to determine ground truth for the labeling process. The tool needed to be able to display video for inspection, segment clips that represent each gesture, and modify the definitions (start time and end time) for each clip.
- **Labeling:** The tool needed to be able to add, delete, or modify labels for each clip. These labels can then be saved to the library.
- **Library management:** The library consisted of many samples which required data management. The tool needed to be able to browse the library, load samples for review, and save out modifications.

The video section of Data Labeler was based on basic video editing programs (such as Kino in Linux) [33]. The user can select clips based on start and stop times that are controlled with sliders. The user can then play the clips, step forward through them, and step backwards through them. This approach allowed me to be very precise in defining sign boundaries and created an easy interface for adjusting and reviewing sign clip definitions.

As each clip is defined, labels are associated with the clip. The interface for labeling allows for labeling with the three criteria used for this study, but the interface can be modified for use with other criteria. Each clip is represented by a tab, which are ordered chronologically by start time. Labels can be reviewed, modified, saved, and deleted.

I built the library management tool using the libraries provided in the GART toolkit, which was developed in previous work [89]. An import tool was constructed to convert the raw data collected in the game into a GART library file. Additional tools were then built that connected the library to the video editing tools and provided cross-indexing of domain-specific information such as video file location and connection to GUI components. For more information on GART see Appendix A.

#### 4.4.2 Data Labeler

Figure 20 shows a screen shot of the labeling tool “Data Labeler.” When a phrase sample is loaded into Data Labeler, all of the information about the sample is loaded, and the video is displayed. The labeling tabs (close-up image shown in top of Figure 22) for an unlabeled sample are initially empty. The sign sample is divided into multiple clips, each of which represent a sign or significant gesture. The video can be played, and then a clip selected, using the video editing tools (close-up image shown in Figure 21).

Once the video clip has been adjusted to select a sign and define the sign boundaries, the clip can be labeled. The user can select the sign definition, dominant hand, and rate the quality of signing. The “Add Label” button will then add the new label to the tabs at the top of the page. The tabs are displayed in order by time stamp. The user can select any tab by clicking on it, and the label can then be reviewed, modified, saved, or deleted, which is very useful for error-checking.

The navigator (close-up image shown in Figure 23) is displayed at the bottom of the program screen and displays information about the sample including the location of the source files and the correct transcription. A drop-down box on the navigator is used to tag the example as correct or incorrect. The navigator also allows the user to navigate the library by selecting samples and saving changes.

*NEW*	sign: start_sentence	sign: white	sign: black	sign: spider	sign: in	sign: white	sign: flowers
0 - 0	quality: good	quality: ok	quality: good	quality: good	quality: good	quality: good	quality: good
	0 - 13	13 - 35	35 - 52	52 - 76	76 - 102	102 - 113	113 - 122

Tag: spider  
Value: good  
Hand: both\_right

Controls

Sign:  ▾

Active hands / Dominant hand:  ▾

Quality of Signing:  ▾

Start: 52

End: 76

Controls

76frame:

/media/disk/COPYCATOFFLINE\_SUMMER2009/DataExtraction/data/S5/Phrase\_6/1

Tag:  ▾

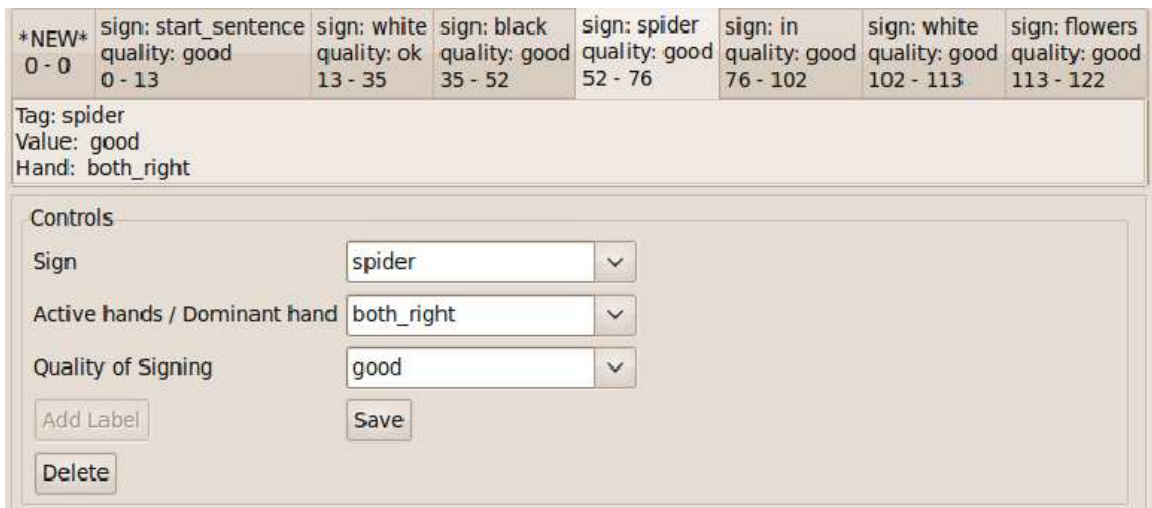
Filename: /media/disk/COPYCATOFFLINE\_SUMMER2009/Features/unnorm/847.txt  
Content: black spider in white flowers

Frames/second: 20  200

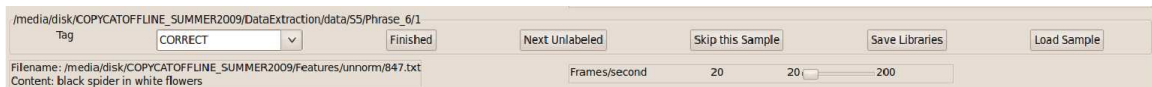
**Figure 20:** Screen shot of the Data Labeler with a labeled sample.



**Figure 21:** Close-up of the Data Labeler video clip editing controls.



**Figure 22:** Close-up of the Data Labeler label editing tabs and controls.



**Figure 23:** Close-up of the Data Labeler sample navigator.

## 4.5 *Labeled Sets*

The labels were used in several different ways to generate data sets to evaluate the impact of different labeling information on modeling. Table 8 shows a comparison of different sets. These permutations included labels for signs, handedness, and quality. Additionally they used two different schemes for segmenting the signs by time. Hand-labeled time segmentation uses the time stamps that were hand labeled with the data. The auto-generated segmentation assumes that the recognizer will converge on label boundaries based on training.

- **Set 0** is a basic transcription of the signs, including out of vocabulary signs. Example: CAT
- **Set 1:** is a transcription of the signs with labels for handedness. Time stamps are hand-labeled. Example: CAT\_LEFT, CAT\_RIGHT, CAT\_BOTH
- **Set 2** is a transcription of the signs with labels for quality. Time stamps are hand labeled. Example: CAT\_GOOD, CAT\_OK, CAT\_BAD
- **Set 3** is a transcription of the signs with labels for handedness and quality. Time stamps are hand labeled. Example: CAT\_LEFT\_GOOD, CAT\_RIGHT\_OK, CAT\_BOTH\_BAD
- **Set 4** is a transcription where all classes are labeled as GARBAGE. Time stamps are hand labeled. It is used to for training various garbage classes for comparison. Set 4 is not included on most charts because it was not used as an evaluation set.
- **Set 5** is a basic transcription of the signs, including out-of-vocabulary signs. Time stamps are auto-generated.
- **Set 6** is a basic transcription of the signs using only game vocabulary. Time stamps are auto-generated.

Auto-generated time labels can be created in HTK by using embedded training. The user can pass a set of sample transcriptions without time stamps and a set of HMM definitions.

**Table 8:** Permutations of labeling information used in evaluations.

Permutation	Includes				Time Segmentation
	Sign Label	Out of Vocabulary	Handedness	Quality	
Set 0	✓	✓			hand labeled
Set 1	✓	✓	✓		hand labeled
Set 2	✓	✓		✓	hand labeled
Set 3	✓	✓	✓	✓	hand labeled
Set 5	✓	✓			auto-generated
Set 6	✓				auto-generated

HTK will concatenate the HMMs for each sample based on the sequence of labels in the transcriptions and run the Forward-Backward Algorithm normally. After all the samples have been processed, the new parameter estimates are generated from the weighted sums of the concatenated models, and HTK outputs an updated HMM set [161].

## 4.6 Analysis of Labeled Data

### 4.6.1 Unique Classes Per Set

Table 9 shows the distribution of unique classes per labeling set. These numbers show the count of labeling permutations represented with each data set. Set #6 is the minimum game vocabulary and has the least number of classes. Set #3 is the most complex labeling set and has the most classes. The higher number of classes will result in few examples per class, but each class will represent a more refined definition.

### 4.6.2 Distribution of Signs Across Students

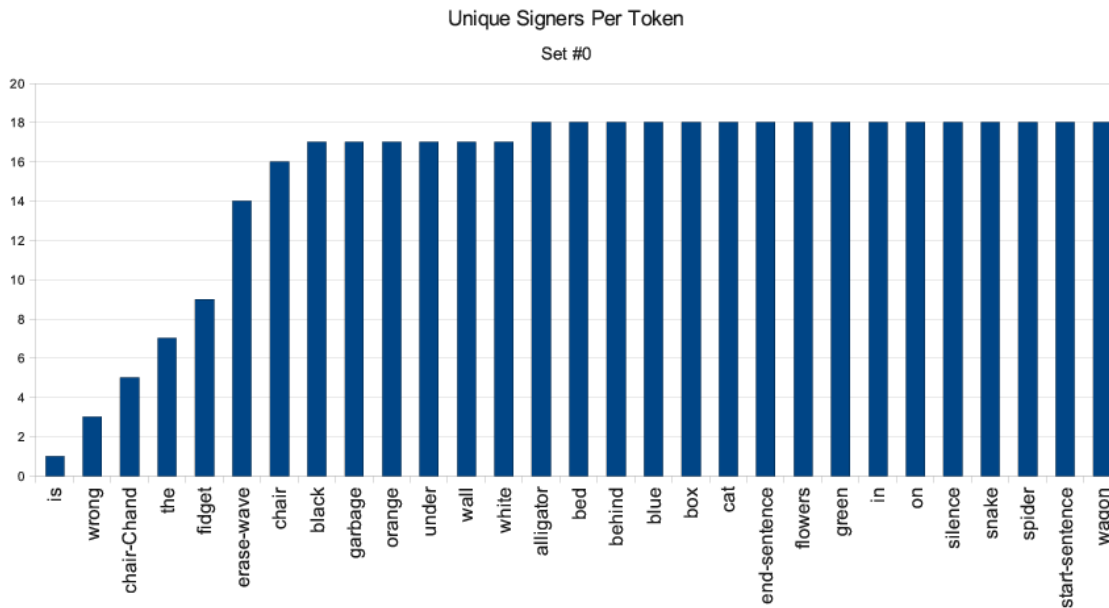
Appendix B shows a full listing of the frequency distributions of labels, per scheme, across the data sets. Figures 24, 25, 26, 27, 28, and 29 are bar charts that show the distributions of unique signers per class for each set. These charts are a helpful aggregate visualization

**Table 9:** Distribution of unique classes per set

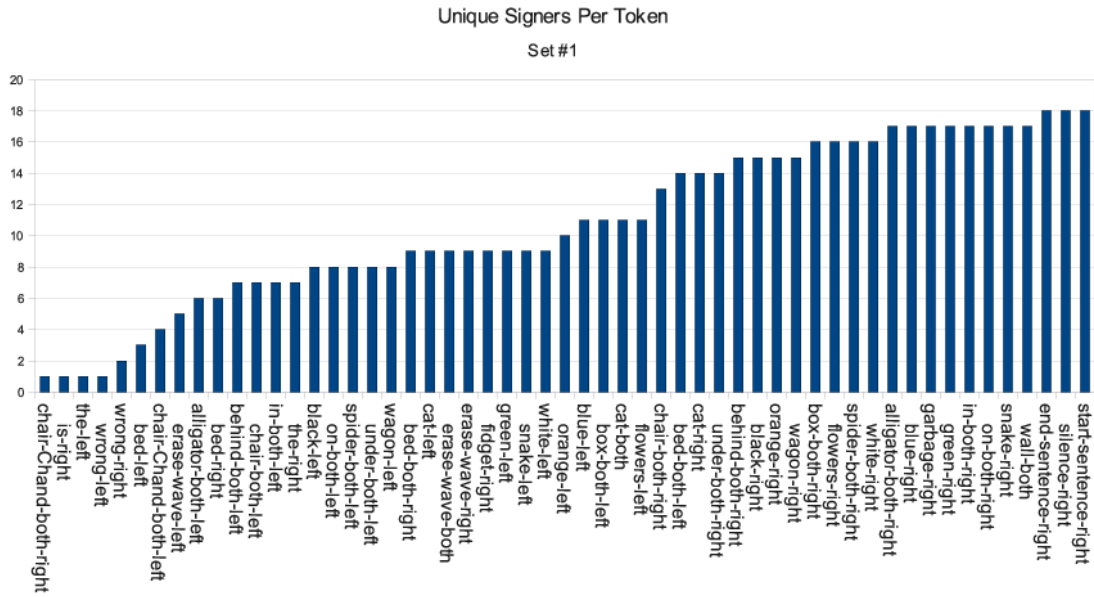
Set	Fall	Spring I	Spring II	All
0	28	27	29	29
1	51	47	54	55
2	60	48	76	76
3	102	79	132	143
5	27	26	28	28
6	21	21	21	21

of the content of the tables in Appendix B. The more specific labeling schemes result in more classes per set. As the number of classes increase, each class tend to have fewer unique signers.

Classes which have too few unique signers or have too few examples were later re-labeled to an appropriate larger grouping for automatic sign recognition. Sometimes these classes can be merged, such as as single BAD example moved into the pool for OK. In cases where there are no related signs, the class may be relabeled as GARBAGE. For example, the sign IS was used by only one student and therefore was relabeled as GARBAGE.



**Figure 24:** Unique signers per class for Set #0 (Shown to visualize trends - not all labels are displayed)



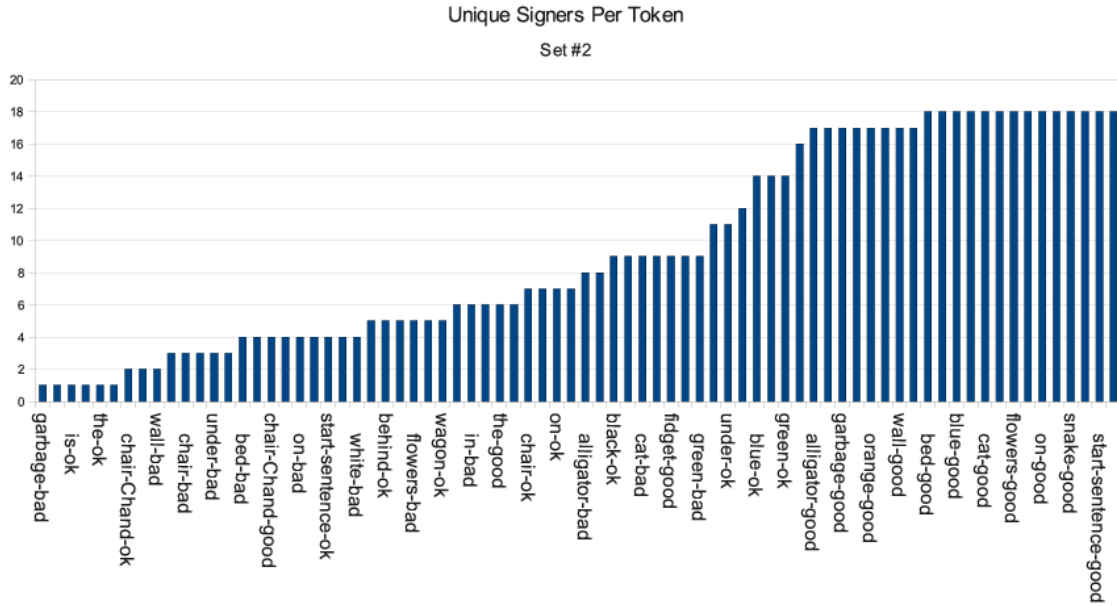
**Figure 25:** Unique signers per class for Set #1 (Shown to visualize trends - not all labels are displayed)

Appendix B, Table 52 illustrates the distribution across Set #0 (shown summarized in Figure 24). Of the 29 classes, 16 of them have examples from every student. One class (IS) has examples from only one student. The sign for IS is a SEE sign that is not frequently used within the schools we visit.

Appendix B, Table 53 illustrates the distribution across Set #1 (shown summarized in Figure 25). In this set, only 3 of the 55 classes have examples from every student. These three signs are all non-vocabulary signs, which do not exhibit handedness: START-SENTENCE, END-SENTENCE, and SILENCE. All of the students used their right hands for the mouse.

Appendix B, Table 54 illustrates the distribution across Set #2, which has 76 classes (shown summarized in Figure 26). In this set only 14 of the classes have examples from every student, but most of the classes have examples from at least half of the students. All of the GOOD examples have examples from over half the students except for CHAIR\_CHAND, IS, THE and WRONG, which are all out-of-vocabulary signs.

Appendix B, Table 55 illustrates the distribution across Set #3, which has 143 classes

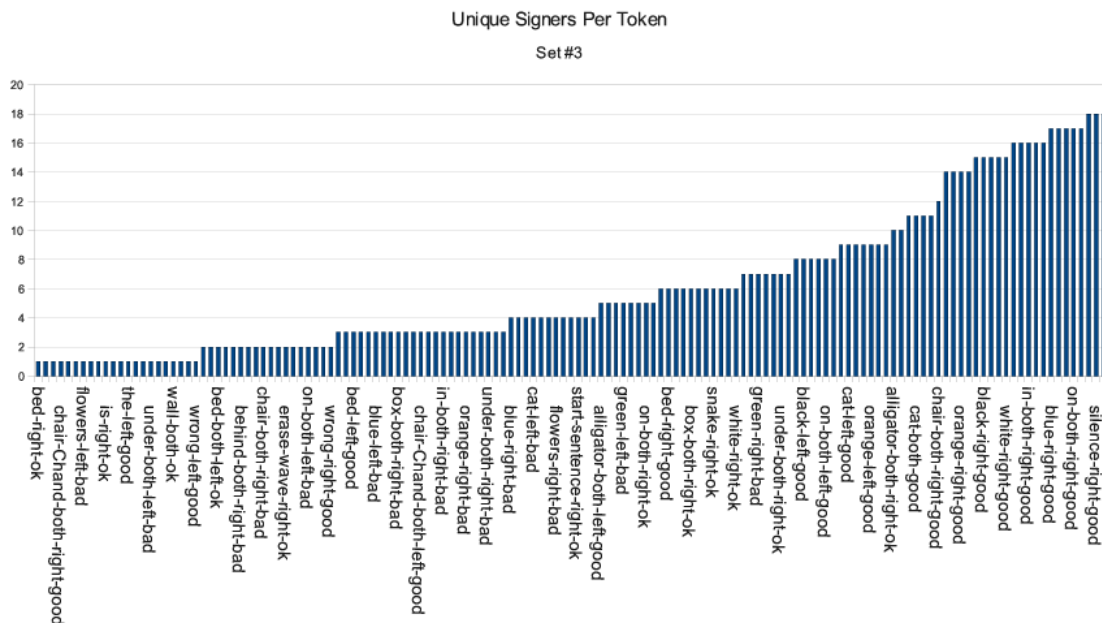


**Figure 26:** Unique signers per class for Set #2 (Shown to visualize trends - not all labels are displayed)

(shown summarized in Figure 27). This set has the highest number of classes and has the steepest curve on its tallies. This set has 22 different labels that only have one unique signer and 18 more labels that have only two unique signers. This set provides the most information per label, but many of these classes were later consolidated in the recognition phase. This consolidation of classes included all of the single signer labels and several of the two signer classes that had too few examples to train accurately. The 143 total labels were later reduced to 120 class classes for use with the automatic recognition.

Appendix B, Table 56 illustrates the distribution across Set #5 (shown summarized in Figure 28). This set contains the same class distribution as Set #0, but uses automatic time segmentation for the automatic recognition phase.

Appendix B, Table 57 illustrates the distribution across Set #6 (shown summarized in Figure 28). This set excludes non-game vocabulary except for the START SENTENCE and END SENTENCE gestures, which reduces the number of classes to 21. Set #6 also uses the automatic time segmentation for the automatic recognition phase (see Chapter 6).



**Figure 27:** Unique signers per class for Set #3 (Shown to visualize trends - not all labels are displayed)

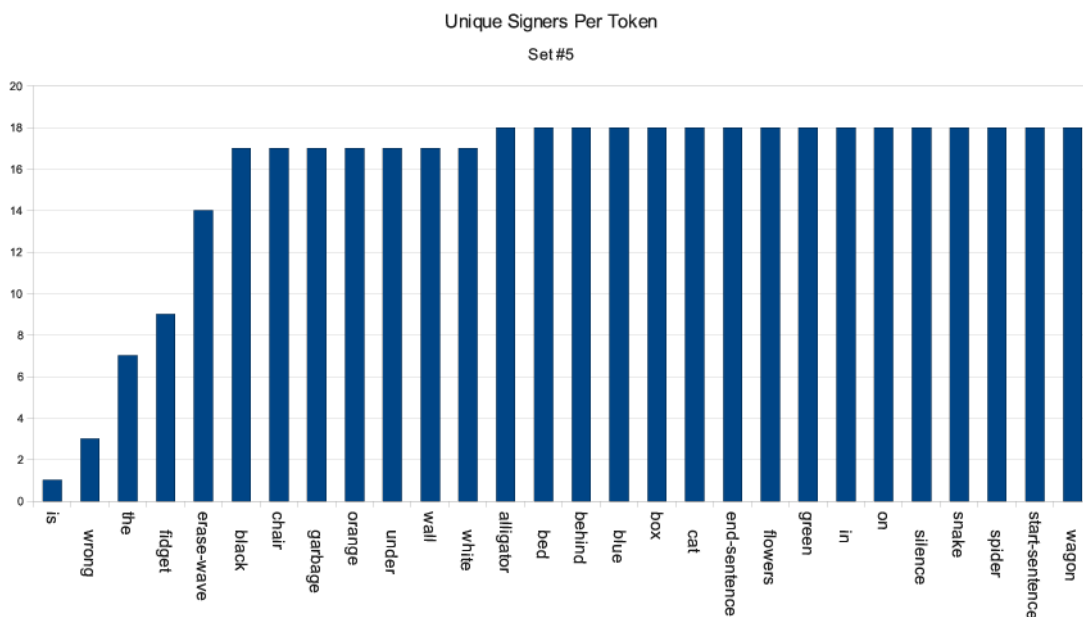
### 4.6.3 Dominant Hand Switching

The children’s signing handedness did not directly correspond to their dominant handedness for other activities and was inconsistent even within the phrases. Dominant hand switching is a symptom of their low fluency and is common among children.

Table 10 shows the handedness variations that occurred for each sign. Signs labeled as BOTH are vocabulary which are symmetrical signs. These signs do not show a dominant hand, but are included in the distribution for completion.

The distribution of handedness for the students varies significantly. Four students (#5, #10 #13, and #17) show a very strong handedness, with only 1% of their signs off-hand dominant. All four of these students showed right hand dominance. In comparison, the student with the strongest left hand dominance (#11) was right hand dominant 35% of the time. Three students (#7, #14, and #15) show a fairly even distribution of dominance and have the dominant hands all split in the 40%-50% range.

Even with the variation in hand dominance, the distribution across all of the students

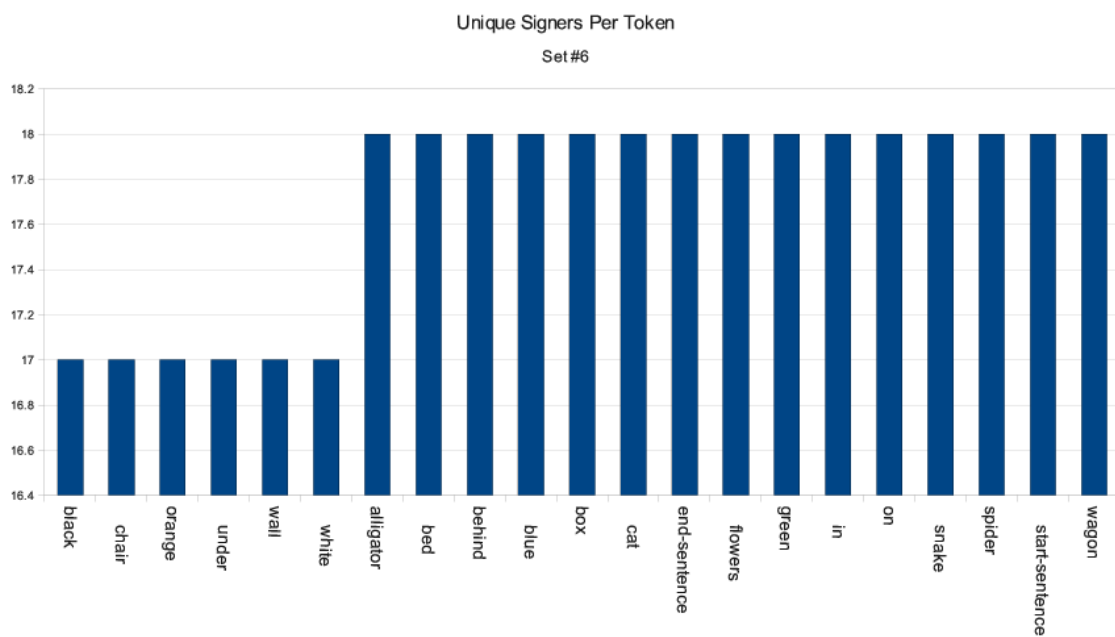


**Figure 28:** Unique signers per class for Set #5 (Shown to visualize trends - not all labels are displayed)

is 79% right hand dominant and 4% left hand dominant. This distribution results in a substantially smaller training set for left-handed signs, which will result in less rigorously trained signs. Though the overall left hand dominance across all signs is only 4%, every student in the study performed at least one left-handed sign. This discrepancy in training data will potentially affect the overwhelming majority of students who play the game.

#### 4.6.4 Quality of Signing

Table 11 shows the variation in quality of sign across the data set. For all students the distribution is 94% GOOD, 4% OK, and 2% BAD. Across all the data sets the lowest performing student (#3) has the distribution of 77% GOOD, 9% OK, and 14% BAD. The best performing student (#15) has the distribution of 99% GOOD, 1% OK, and 0% BAD.



**Figure 29:** Unique signers per class for Set #6 (Shown to visualize trends - not all labels are displayed)

**Table 10:** Distribution of dominant hands in student signing: Percentages show the portion of signs performed as right hand dominant, left hand dominant, or both (symmetrical signs which do not have a dominant hand).

User	Fall			Spring I			Spring II			All		
	Right	Both	Left	Right	Both	Left	Right	Both	Left	Right	Both	Left
0							95	2	3	95	2	3
1							91	2	7	91	2	7
2	92	2	6	92	5	3				92	3	5
3							59	1	40	59	1	40
4							93	4	3	93	4	3
5	94	6	1	94	5	1				94	5	1
6							85	12	3	85	12	3
7							45	3	52	45	3	52
8							90	5	5	90	5	5
9							91	2	7	91	2	7
10							92	6	1	92	6	1
11	34	4	62	37	4	59	25	0	75	35	4	61
12	93	4	3	92	7	2				92	6	2
13	92	7	1	91	8	0				92	8	1
14							50	2	48	50	2	48
15							43	5	52	43	5	52
16							75	2	23	75	2	23
17							94	5	1	94	5	1
<b>Total</b>	<b>81</b>	<b>5</b>	<b>14</b>	<b>83</b>	<b>6</b>	<b>12</b>	<b>76</b>	<b>4</b>	<b>21</b>	<b>79</b>	<b>4</b>	<b>17</b>

**Table 11:** Breakdown of quality in student signing

User	Fall			Spring I			Spring II			All		
	GOOD	OK	BAD	GOOD	OK	BAD	GOOD	OK	BAD	GOOD	OK	BAD
0							91	9	0	91	9	0
1							84	11	5	84	11	5
2	97	2	1	96	4	0				97	3	0
3							77	9	14	77	9	14
4							97	2	0	97	2	0
5	97	3	1	98	1	1				97	2	1
6							96	4	0	96	4	0
7							89	6	4	89	6	4
8							99	0	1	99	0	1
9							90	8	2	90	8	2
10							96	4	0	96	4	0
11	92	5	3	93	3	3	88	13	0	92	5	3
12	98	1	1	98	1	1				98	1	1
13	95	3	3	97	2	1				95	3	2
14							94	3	3	94	3	3
15							99	1	0	99	1	0
16							92	5	2	92	5	2
17							88	7	5	88	7	5
<b>Total</b>	<b>95</b>	<b>3</b>	<b>2</b>	<b>96</b>	<b>2</b>	<b>1</b>	<b>92</b>	<b>5</b>	<b>2</b>	<b>94</b>	<b>4</b>	<b>2</b>

#### 4.7 Structure of Signed Examples

Table 12 shows a breakdown of the data set content by matches to the game grammar. The table was compiled by comparing the correct sign sequence for a phrase with the transcript of the labeled phrases. Phrase matches were matched for game correctness. Grammar matches were matched to the game grammar of

*[Adjective]SubjectPreposition[Adjective2]Object.*

These examples were examined by their part of speech and not error checked for game correctness. For example, a phrase “GREEN ALLIGATOR ON BLUE WALL” would match the game grammar, but if the wall in question was orange, it would not be an exact phrase match.

Table 12 also shows two groups that use different ways to match the labels. *Game* label lines were matched by using only the game vocabulary and ignoring other labels, while the *All* label lines compared all signs from a transcript. The phrase “GREEN FIDGET ALLIGATOR ON BLUE WALL” would be a match for the *Game* label group since the sign FIDGET would be ignored, but it would not be a match for the *All* label group since FIDGET is not in the grammar.

These numbers are interesting when we begin to look at potential grammars to use in recognition. Grammars are commonly used in speech recognition [68] and previously, we have used both statistical and rule-based, part-of-speech grammars to aid automatic sign recognition [17, 18]. A recognition grammar that matched the game phrase exactly would only match 67.33% of the examples. If the recognition grammar was strictly based on game vocabulary, that number would drop to 43.07% when all labels were used. Table 12 shows that when using labels that are strictly game vocabulary only 71.37% of the samples follow the game grammar, and when using all of the sign labels that number drops to 44.16% .

An examination of the transcripts of the children’s signing shows that there are 428 different permutations of the game grammar when out-of-vocabulary signs are included. If one re-labels all of the non-sign gestures there are 358 permutations. These large numbers of unique permutations of vocabulary over a phrase set of 1191 samples are indicators that

**Table 12:** Analysis of the labeled data for game grammar matching. For the column Label: *Game* indicates that non-game vocabulary signs are ignored and *All* indicates that all signs are used for comparison. For the column Match: *Phrase* indicates a game correct transcription and *Grammar* indicates a transcript match to the generalized game grammar, but not specifically to the intended phrase for the game scenario.

		<b>Three</b>	<b>Four</b>	<b>Five</b>		
<b>Labels</b>	<b>Match</b>	<b>Sign</b>	<b>Sign</b>	<b>Sign</b>	<b>All</b>	<b>Percentage</b>
Game	Phrase	156	132	514	802	67.33%
Game	Grammar	160	165	525	850	71.37%
All	Phrase	96	83	334	513	43.07%
All	Grammar	98	87	341	526	44.16%

a statistical grammar would behave poorly.

These poor indicators for both defined and structural grammars are a new facet of our data set. Our previous data sets have been well structured either due to scripting [17] or due to pruning for errors [18, 163, 164]. This lack of consistent structure means that development of the automatic recognition for live game play should be flexible enough in its structuring to allow for these variations in grammar structure.

#### 4.8 Summary

In this chapter we have described the development of a labeling ontology. I was able to use the ontology to characterize the content of the data set, which increases our understanding of what the children are doing in the samples, as well as how that might affect the automatic sign recognition process. This ontology was used to create six different labeling schemes that will be used in the automatic sign recognition phase. I have examined the frequency of labels, number of unique signers per label, quality content, handedness content, and grammatical structure of the samples in our set. This information will be used for the experimental process described in the next chapter, which will cover automatic recognition.

## CHAPTER V

### MODELING

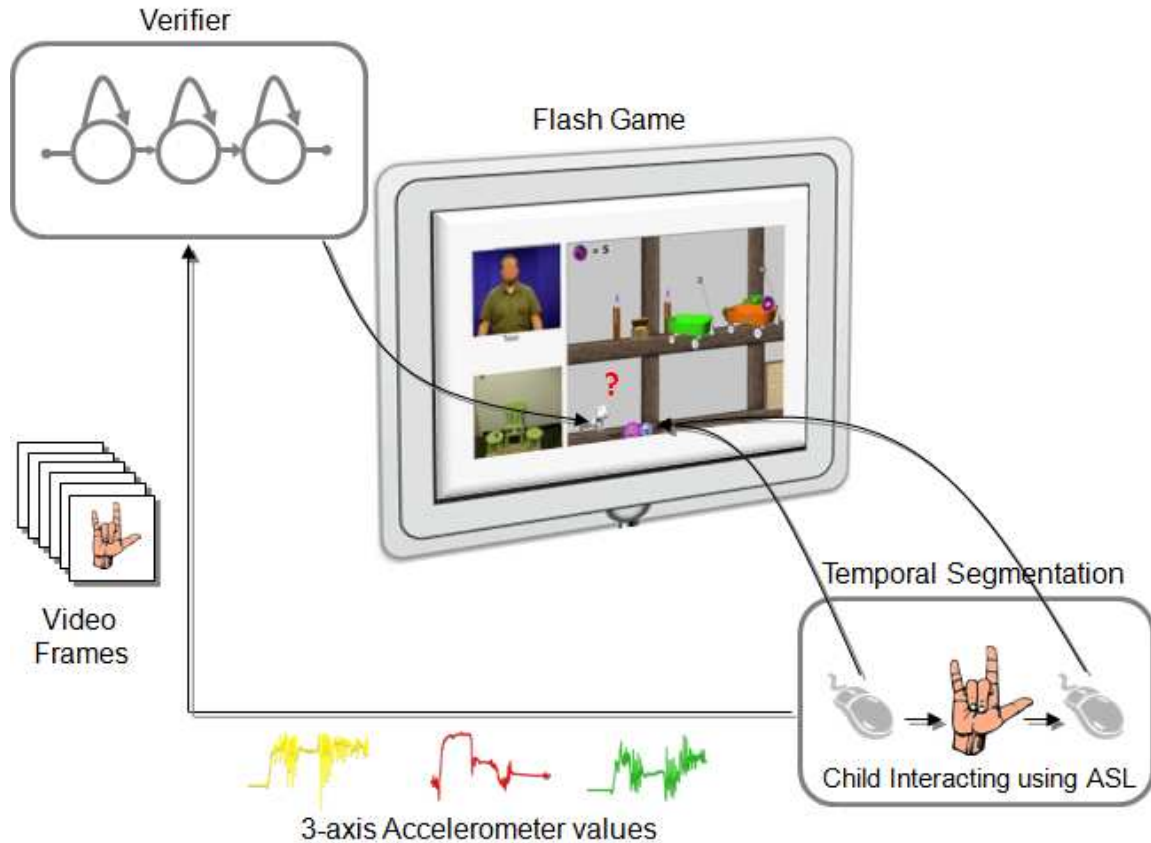
In this chapter I will describe the generation of feature vectors and the use of models for recognition. Our process to generate feature vectors for automatic sign recognition has evolved as a collaboration of multiple researchers in the CopyCat group, over a series of automatic sign recognition projects [130, 17, 95, 18, 163, 164]. The process that I present in this chapter represents the feature vectors that we are currently using in the project. These are the feature vectors that I used in the recognition experiments in Chapter 6. Likewise, the recognition infrastructure has also evolved through a series of projects [17, 151, 95, 89]. Additionally, I have made some customizations beyond our publicly available toolkits for my experiments.

#### *5.1 Data Processing*

CopyCat uses a main control system behind the interface to synchronize data streams from video and accelerometers, along with game play information. All of the data is archived to the computer during both Wizard of Oz game play and live game play. During live game play, these raw data streams are then processed into feature vectors which are passed to the hidden Markov models for recognition. My experiments use the same feature vector processing even though they are performed off-line on archived data from the second deployment. The CopyCat control system and feature generation has been developed as a collaboration of the CopyCat research group [18, 58, 163, 164].

##### **5.1.1 Accelerometers**

Our Bluetooth accelerometers were designed in-house for use with the CopyCat project [17, 18, 150]. The accelerometers are sampled at 40 Hz and they measure a range of +2g to -2g. The game uses two wrist-mounted accelerometers to measure movement of the hands through the signing space. The raw accelerometer data is used to calculate x, y, and z



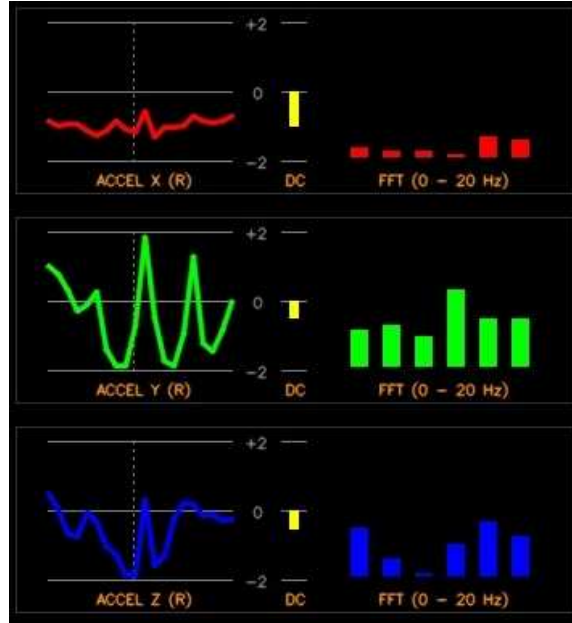
**Figure 30:** Control system for the CopyCat game. The push-to-sign functionality segments the data. The control system synchronizes data streams from video and accelerometers, archives the raw data, and generates feature vectors for each signing segment. The control system also passes game information and the feature vectors to the automatic game, which returns a classification of the phrase as correct or incorrect.

acceleration values, as well as the frequency domain representation of each axis.

### 5.1.2 Image Processing

The video stream for CopyCat is collected from a camcorder which produces 720x480 video frames sampled at 20 frames per second (fps). The images are then converted to HSV color space, and the hands are tracked by hue. To play the game, the child wears gloves of two different colors on their hands. These gloves help enable hand tracking even when hands overlap each other or the face. Head tracking is used to track the child's eyes while they play the game. This video information is used to calculate information on

- Hand detection: Second moment shape descriptors of the hand shapes extracted from



**Figure 31:** Visualization of the FFT for a single three-axis accelerometer

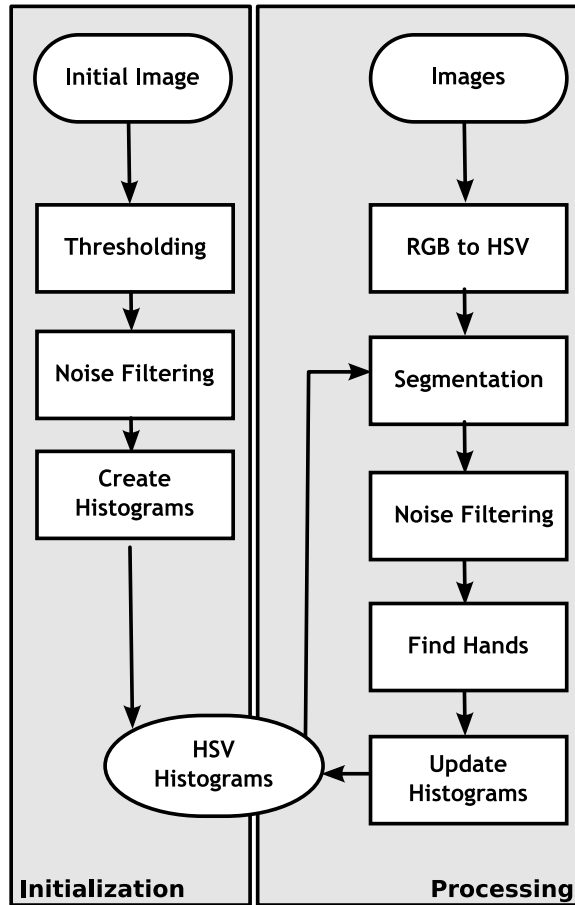
the image (length of major and minor axes, eccentricity, and orientation of major axis)

- Hand shape: Shading-based features obtained by performing PCA on concatenated histograms of V (from HSV) from a 4x4 grid of the extracted hand region
- Hand tracking: Change in location of the hand through the image ( $\Delta x$ ,  $\Delta y$ )
- Head detection: Location of the head in the image and position of the eyes
- Head pose angle: The angle formed between the hand shape center and the horizontal passing through the midpoint between the eyes

### 5.1.3 Hand Tracking

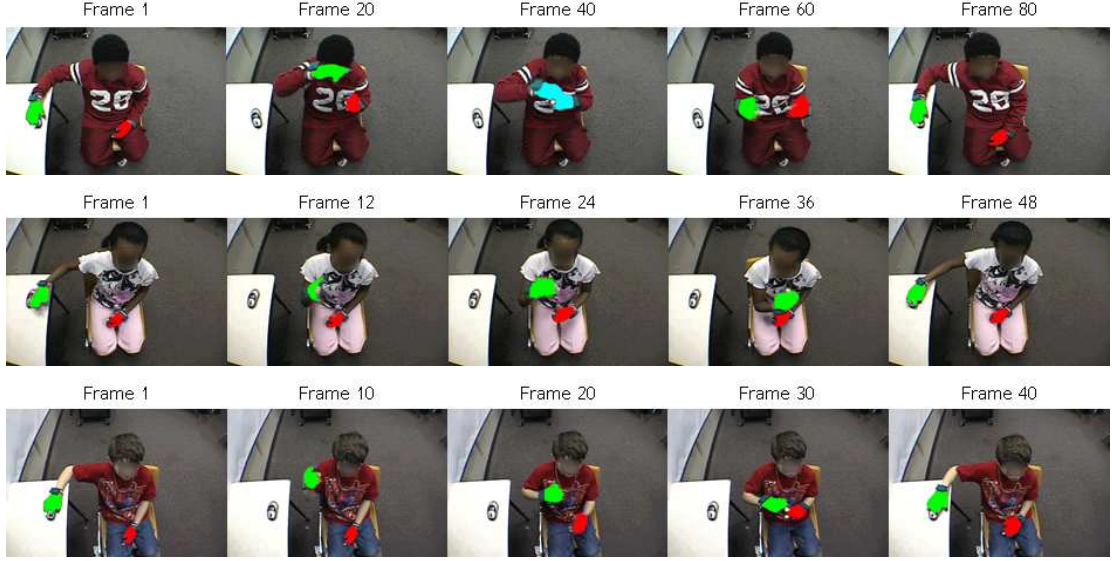
In our system, we require the children to wear colored gloves. These bright colors are easily identified by a computer vision algorithm. Tracking skin tones can be particularly problematic for computer vision in unconstrained environments. Additionally, it is difficult to distinguish when the hands perform signs near the face. Many algorithms have been suggested to segment hand region robustly, even under illumination change [104, 155, 133, 134, 126]. However, some of them address only a narrow range of illumination change, and

some results do not guarantee real-time processing (at least 10 fps with  $720 \times 480$  sized images in our system) or robustness for long image sequences of gestures. Some methods extract similar color regions as well as hand color regions, and the performance strongly depends on the result in the first image frame.



**Figure 32:** Hand segmentation process

In our approach, the image pixel data is converted to HSV color space and used to create histograms for segmentation of the hand region and background, as shown in Figure 32. HSV histograms are used to produce a binary mask using a Bayes classifier [126], and noise is removed by morphological filters including size filtering and hole filtering. The position of the desk and the colored gloves provide a significant marker for starting the gesture recognition; the light color of the desk provides a high contrast environment. The children click the mouse to start and end each phrase, which provides both location and color cues,



**Figure 33:** Segmentation of hands from video clips of the Phase One deployment

as well as a start and end gesture. From these cues, we can successfully extract the mouse hand region for the first frame of the image sequence by simply applying a threshold.

We initially use the hand segmentation to create the starting histogram. Each frame is a segmentation cycle, which provides feedback to the system and helps enhance the discrimination of the color models. HSV histograms are updated with a weight value  $\omega$ , ( $0 < \omega < 1$ ), based on the obtained mask, and then the histograms are normalized

$$H \leftarrow (1 - \omega)H + \omega H^{new}$$

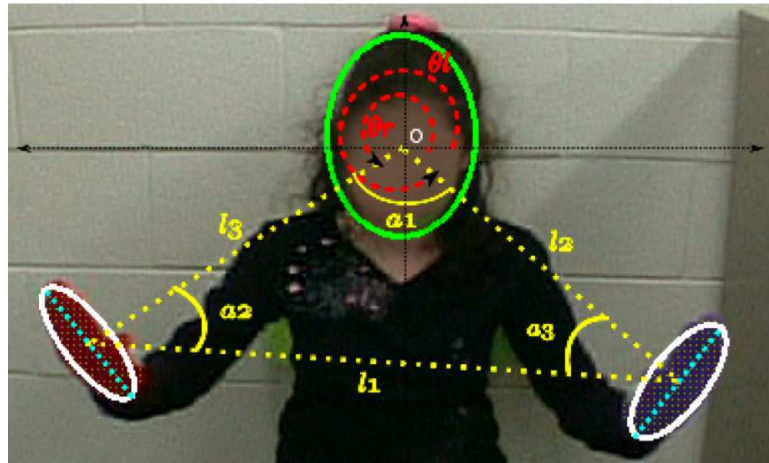
where  $H$  denotes the histogram value for each bin [126].

Figure 32 shows the hand tracking process for later frames. The segmentation of the hand region and the update of HSV histograms are the same as the procedure in the first frame. To find both hands in the binary mask, we consider the size of hand shapes and the distance between the center position of the candidate blobs, as well as the hand positions in the previous image frame.

Figure 33 shows the results of the image processing for several image sequences. Processing occurs at  $48.574ms/frame$  (20.59 fps) in a laptop computer with a  $1GHz$  processor. We found that the tracking results were acceptable, even when the child wears a shirt with color patterns similar to the gloves.

#### 5.1.4 Head Tracking

Head tracking was implemented in OpenCV [16]. We used boosted classifiers which were trained with the Haar training utility [16]. The head tracking training set contained 800 positive examples of the children’s heads (both with and without hand occlusion) and 400 negative examples selected from our images. The tracker was configured to output the largest object found in the region of interest, which increased efficiency [164].



**Figure 34:** Image of child signing which has been annotated with tracking information for the head and hands

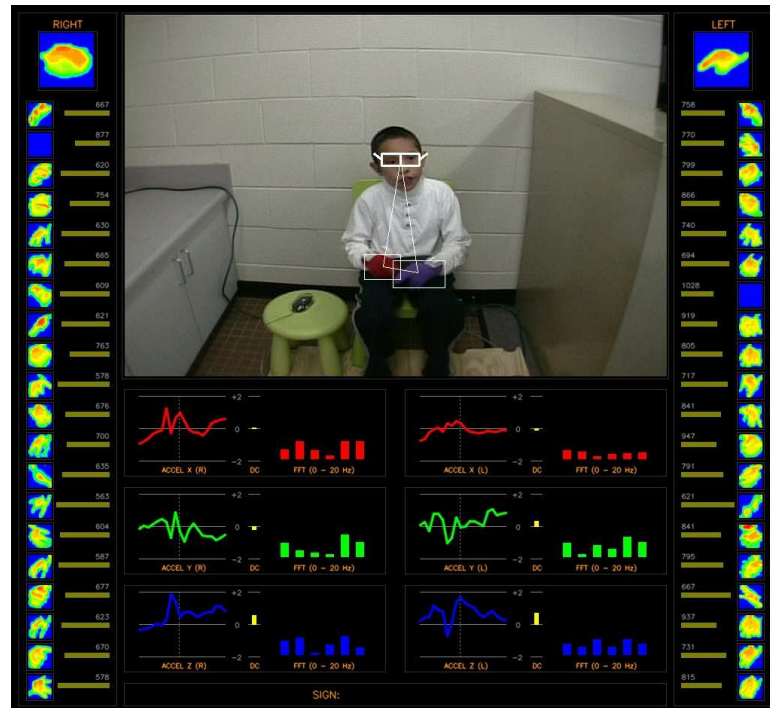
Figure 34 shows a visualization of the tracking information. The variables are defined as follows: the head is  $O$ , the right hand is  $R$ , and the left hand is  $L$ . The angles  $a_1$ ,  $a_2$ , and  $a_3$  are formed by the triangle defined by  $ORL$ . The variables  $l_1$ ,  $l_2$ , and  $l_3$  are the normalized length of the triangle  $ORL$ . The angles  $\theta_l$  and  $\theta_r$  are defined by the angle between the line  $OL$  and the major axis of the left hand, and the angle between the line  $OR$  and the major axis of the right hand, respectively.

#### 5.1.5 Feature Vectors

Our feature vector is composed of information from both the video feed and the accelerometers. Table 5.1.5 shows a summary of the features, listed by type groupings. Figure 35 shows a screen shot of a visualization tool for displaying the information from the feature vectors.

**Table 13:** Feature vector description

Type	Description
Blob	second moment shape descriptors (length of major and minor axes, eccentricity, orientation of major axis)
Hand Shape	shading based features obtained by performing PCA on concatenated histograms of V (from HSV) from a 4x4 grid of the extracted hand region.
2D Image Motion	$dx$ and $dy$ of the blob center
Acceleration	x, y & z acceleration values and frequency domain representation of each axis.
Pose (2D geometry)	angle formed between the blob center and the horizontal passing through the midpoint between the eyes



**Figure 35:** Screen shot from the verification tool (Image used courtesy of Zahoor Zafrulla)

## **5.2 Recognition Experiments**

The recognition experiments are defined by how the data set is divided up and used in the testing and training process. The data is divided into training sets, which are the examples used to train the models, and testing sets, which are the examples used for the recognition tests.

### **5.2.1 Testing on Training**

Testing on training experiments are frequently used to assess a recognition system quickly. The entire data set is used as both the training and validation (testing) set. This redundancy means that the models will be tested using only data that is used to build them. This approach gives us an idea of the upper bounds of recognition performance in the data set.

### **5.2.2 Leave-one-out**

For my “leave-one-out” experiments in Chapter 6, I use m-fold cross validation where the sets are grouped by child. Each validation set represents a single child’s game play data. The training set is then composed of the data from all other children’s data. This approach allows me to evaluate how well the system responds to use by a new child, which is an important metric as we design the system for larger scale deployment. By testing against each child’s data, we can also examine how the system responds to each of the different children.

### **5.2.3 Data Divisions**

Table 14 shows the distribution of phrases and individual signs by user. The variation in the number of signs and phrases collected is due to the fact that children participated in one, two, or three of the testing periods. Students who participated in more testing periods tend to have more data. Additionally, the number of phrases completed per session varied by child depending on how quickly they played the game.

The wide variance in sample size per child is worth noting, because it affects both the testing on training and leave-one-out experiments. Classically, m-fold cross validation uses equal-sized sets [31], but our m-fold validation is grouped by child. This grouping means

**Table 14:** Distribution of samples by user

	<b>Phrase</b>	<b>Sign</b>
<b>User</b>	<b>Examples</b>	<b>Examples</b>
0	19	159
1	44	334
2	139	1043
3	15	121
4	52	338
5	134	1059
6	14	94
7	39	279
8	20	130
9	76	488
10	59	394
11	117	934
12	92	665
13	130	1169
14	55	512
15	59	392
16	77	575
17	50	369
Total	1191	9055

that for the cross validation there is a variance in the size of the sets used for testing and training. The size of the validation set varies from 1.04% of the data (94 signs from User #6 out of 9055 total) to 12.91% of the data (1169 signs from User #13). Likewise, the children’s sample representation within the entire set for testing on training is disproportionate. As the size of the CopyCat data set grows with future deployments, the impact of this variation should lessen as long as new children are being continuously added.

#### 5.2.4 Recognition Metrics

Each experimental run results in a series of metrics that evaluate how well the recognition results match the ground truth labeled data. We will use standard word level speech recognition metrics for measuring the performance of our recognition. The following symbols are defined

- **H** is the number of correct instances
- **D** is the number of deletion errors
- **S** is the number of substitution errors

**Table 15:** Definition of variables used to calculate recognition metrics

Var	Definition	Example: <i>ALLIGATOR ON WALL</i>
<b>H</b>	Hits - Correct matches	H=3 <i>ALLIGATOR ON WALL</i> H=2 <i>ALLIGATOR ON CHAIR</i>
<b>D</b>	Deletion errors	D=1 <i>ALLIGATOR WALL</i> D=2 <i>ALLIGATOR</i>
<b>S</b>	Substitution errors	S=1 <i>SPIDER ON WALL</i> S=2 <i>SPIDER ON CHAIR</i>
<b>I</b>	Insertion errors	I=1 <i>ALLIGATOR UNDER ON WALL</i> I=2 <i>ALLIGATOR ALLIGATOR UNDER ON WALL</i>
<b>N</b>	Total number of samples	N=3 <i>ALLIGATOR ON WALL</i>

- **I** is the number of insertion errors
- **N** is the total number of instances

Word Correctness is calculated by:  $Correct = \frac{H}{N}x100\%$  where N is the number of instances by sign. Word Accuracy is calculated by:  $Correct = \frac{H-I}{N}x100\%$  where N is the number of instances by sign. Sentence Correctness is calculated by:  $Correct = \frac{H}{N}x100\%$  where N is the number of instances by phrase. Table 15 shows a breakdown of the numbers used to calculate the metrics, along with examples of how they are calculated.

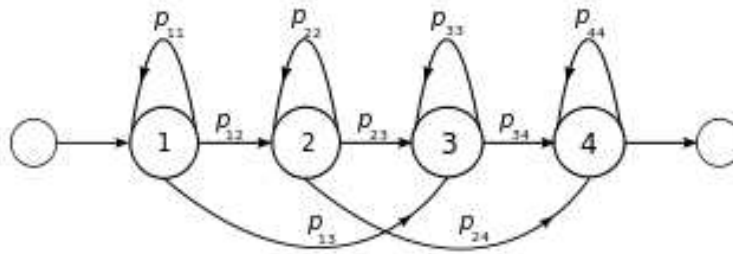
### 5.3 Recognition Infrastructure

Hidden Markov models (HMMs) are stochastic models that represent unknown processes as a series of observations. In previous work we have had success modeling ASL with HMMs [17, 151, 95, 18, 89, 163, 164]. We previously designed two toolkits, GT<sup>2</sup>k [151, 95, 18] and GART [89], that leverage the speech-based tools in Cambridge University’s Hidden Markov Model Toolkit (HTK) [161]. The recognition infrastructure used for the experiments in Chapter 6 was built using a combination of these tools, as well as some additional custom tools.

#### 5.3.1 Hidden Markov Models

The HMM topology for use with CopyCat was experimentally determined. A visualization of the topology is shown in Figure 36. The HMM is a left-to-right model with a single skip transition used for each state. We use a continuous density HMM with 6 states, 4 of which

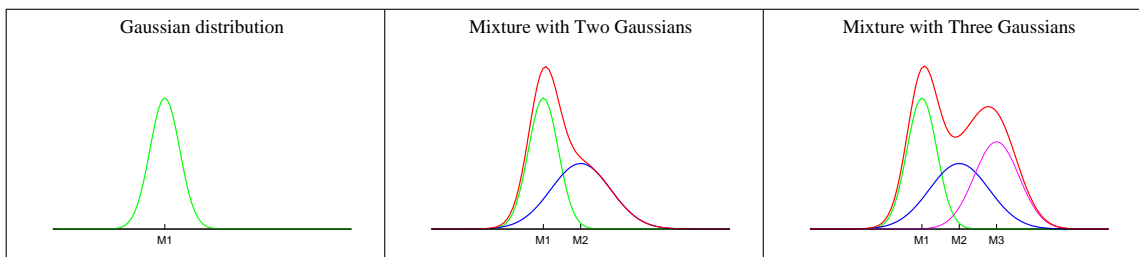
are emitting states. The emitting states have three component mixture Gaussians. Full covariance matrices can be appropriate for very large data sets, but the CopyCat data set is not yet large enough, so we use a diagonal covariance matrix [161].



**Figure 36:** Visualization of a four state left to right HMM with two skip states

All of the classes in the testing on training experiments were trained with three component mixture Gaussians. These mixture models add a mathematical flexibility to our modeling that allows for more precise models. Figure 37 shows Gaussians of one, two, and three components; the more components in the mixture, the more closely a complicated sample distribution can be modeled. One significant danger of mixture models is over-fitting to data [31]. The component number of three was experimentally determined. Testing evaluated improvement in modeling accuracies, avoidance of over-fitting, and sufficient sample coverage.

Though the three component models were found to be optimal for our testing overall, the leave-one-out by child experiments frequently divided up our samples in uneven ways. As discussed in Chapter 4, some label classes had few unique signers. These classes were susceptible to a disproportionately reduced sample size during the leave-one-out by child



**Figure 37:** Mixture of Gaussians. From left to right: Single Gaussian, Mixture with two components, Mixture with three components

experiments. Some of these classes did not have enough data or variation to train with three mixtures, so either two component mixtures were used or a single Gaussian. A listing of the modified models are listed below:

- **Set 1:** WRONG\_RIGHT
- **Set 2:** END\_SENTENCE\_BAD, CHAIR\_CHAND\_OK
- **Set 3:** ALLIGATOR\_BOTH\_LEFT\_OK, BED\_BOTH\_LEFT\_BAD, BED\_BOTH\_LEFT\_OK, BED\_BOTH\_RIGHT\_BAD, BEHIND\_BOTH\_LEFT\_BAD, BEHIND\_BOTH\_RIGHT\_OK, BLACK\_RIGHT\_BAD, CHAIR\_CHAND\_BOTH\_LEFT\_OK, END\_SENTENCE\_RIGHT\_BAD, FLOWERS\_LEFT\_OK, ON\_BOTH\_LEFT\_BAD, ON\_BOTH\_LEFT\_OK, SPIDER\_BOTH\_LEFT\_GOOD, WRONG\_RIGHT\_GOOD

It is worth noting that all but one of the classes on this list have three or fewer unique signers and ten or fewer examples. It is not surprising that these classes would lack the variance needed to fit a multiple component mixture model, since they have both a small number of samples and a small number of unique signers. Only SPIDER\_BOTH\_LEFT\_GOOD, with 83 examples and 8 unique signers, had difficulties with insufficient variance for a larger number of samples.

### 5.3.2 HTK

The Hidden Markov Model Toolkit (HTK) is a publicly available toolkit for modeling hidden Markov models [161]. HTK provides HMMs in the context of a language infrastructure for use in speech recognition. HTK is designed as a speech recognition toolkit, but the core of HTK can be used to build hidden Markov models for any time series. There are two main classes of tools in HTK: those that build models and estimate the parameters from training samples and those that are used to provide transcriptions for unknown samples. Though much of the infrastructure of HTK is geared towards speech recognition, we have found that the toolkit can be used very effectively for other recognition tasks [89]. Appendix A has more details on the structure and use of HTK.

### 5.3.3 GT<sup>2</sup>k

In my previous work we developed the Georgia Tech Gesture Toolkit (GT<sup>2</sup>k) which encompasses a set of tools designed to allow easy development of gesture recognition components of larger systems using HTK [151]. GT<sup>2</sup>k is a publicly available toolkit for developing gesture-based recognition systems. The toolkit provides capabilities for training models and allows for both real-time and off-line recognition. Appendix A has more details on the structure and use of GT<sup>2</sup>k.

### 5.3.4 GART

In my previous work we developed the **G**esture and **A**ctivity **R**ecognition **T**oolit (GART), which is a Java-based user interface toolkit designed to enable the development of gesture-based applications [89]. GART provides an abstraction to machine learning algorithms suitable for modeling and recognizing different types of gestures, as well as support for the data collection and the training process. Appendix A has more details on the structure and use of GART.

Although GART and GT<sup>2</sup>k are both tools built on top of HTK, they are designed to be fundamentally different. GT<sup>2</sup>k is a set of scripts and tools provided to perform batch experiments in HTK. These scripts help organize experiments and aid in generating configuration files. GT<sup>2</sup>k was predominantly used to script batch experimental runs and to collate results. GART is a Java-based library which was designed to create gesture-based user interfaces. GART makes direct calls to HTK for training models and live recognition, but the bulk of the GART infrastructure is in tools to aid application programmers. GART was predominantly used for library management in the development of the labeling tool.

### 5.3.5 Building the Infrastructure

The recognition infrastructure was designed for the following data flow: label samples, train models using samples, and test model performance using samples. Each of the tools described contributes to the process in the following way:

- **GART, Data Labeler, Customized Programs:** Data is labeled, and the labels are used to output HTK format files. The Data Labeler (described in Chapter 4) is built using the GART libraries. It loads the samples and displays them for annotation. After the annotation is complete, the sample information is saved to a GART Library. GART's library utilities are combined with a custom program which outputs the data in HTK format files for all of the labeling permutations.
- **GT<sup>2</sup>k, Customized Scripts:** The experiments are defined and configured for batch runs. The scripts and tools from GT<sup>2</sup>k are designed to simplify the process of designing experiments in HTK. These are customized for each experiment definition.
- **HTK, GT<sup>2</sup>k, Customized Scripts:** The experiments are run and the output is parsed for analysis. In addition to the experiments described in the next chapter, several experiments were run to establish configurations for HMM topologies, mixture models, and other HTK parameters.

#### 5.4 *Summary*

In order to run the experiments in the next chapter, we must generate feature vectors for our samples and build the infrastructure used to train and test our models. We combine information gathered from the wrist-mounted accelerometers, hand tracking, and head tracking to create feature vectors for each sample. The systems and techniques that generate these feature vectors have been a product of the evolution of the CopyCat project and a collaboration of the CopyCat research group.

Our recognition experiments are defined by the distribution of data for training and testing, as well as the labeling schemes defined in Chapter 4. The infrastructure for our recognition experiments was built with the aid of three tools: HTK, GT<sup>2</sup>k, and GART. Two of these tools, GT<sup>2</sup>k, and GART, are publicly available toolkits I helped build.

## CHAPTER VI

### EXPERIMENTS

Once the labeling phase was complete, the labels were then used in a series of exploratory recognition experiments. These experiments are a progression leading towards my recognition engine. My experiments are presented in the order that they were done leading to the final recognition engine. This discussion covers the progression of recognition experiments that results in the final automatic sign language recognition engine used in the game.

#### *6.1 Research Goals*

The work covered in this chapter can be summarized by

- Contributions:
  - I will use the ontology to improve automatic sign language recognition for the CopyCat game.
- Research Questions:
  - Can we use these labels to improve our automatic sign language recognition?
  - How do variations in the application of the labeling information affect recognition results?
- Hypotheses:
  - More detailed labeling will improve recognition.
  - Handedness and sign variations will be the most important labeling details.
  - Quality may allow us to create models that accept varied fluency in signing.
- Method:
  - Use hand-labeled transcripts to create permutations of labeling schemes for recognition models
- Data Collected:

- Train models using these labeling schemes.
- Run recognition tests on the different model sets
- Analysis:
  - Look for performance trends in recognition for the various schemes.
  - Determine best approaches to optimize the recognition engine for the game.

## 6.2 *Leave-one-out Tests*

As discussed in Chapter 5, the leave-one-out tests were run by separating the data into groups based on the signer. Through these experiments, each child’s data was used as a test set, while the rest of the data was used as the training set.

### 6.2.1 Experimental Sets

Each experiment was defined by its labeling scheme. The labeling scheme sets are defined from the results of the labeling phase described in Chapter 4. The sets are as follows

**Set #0** used a labeling scheme which consisted of mapping each sign to its vocabulary label and included out of vocabulary signs. The time segmentations (sign boundaries) were hand labeled by visual inspection.

**Set #1** used a labeling scheme which consisted of mapping each sign to a label composed of the vocabulary word and handedness information, and included out of vocabulary signs. The time segmentations (sign boundaries) were hand labeled by visual inspection.

**Set #2** used a labeling scheme which consisted of mapping each sign to a label composed of the vocabulary word and quality information, and included out of vocabulary signs. The time segmentations (sign boundaries) were hand labeled by visual inspection.

**Set #3** used a labeling scheme which consisted of mapping each sign to a label composed of the vocabulary word and both handedness and quality information, and included out of vocabulary signs. The time segmentations (sign boundaries) were hand labeled by visual inspection.

**Set #4** is a set which included all signs in a generic sign class. This set was not used in this round of experimentation, but will be used in later experiments on the garbage class

**Table 16:** Percentage word accuracy for per student, per scheme.

Child	Set #0	Set #1	Set #2	Set #3	Set #5	Set #6	Avg
0	32.70	33.33	35.85	32.08	30.82	30.40	32.53
1	54.19	56.89	47.31	46.71	48.20	50.82	50.69
2	60.59	65.00	57.91	62.90	59.06	63.94	61.57
3	29.75	36.36	23.97	25.62	27.27	25.96	28.16
4	73.96	78.70	71.01	74.85	72.49	67.56	73.09
5	49.86	50.52	43.44	50.99	45.51	44.07	47.40
6	51.06	59.57	50.00	57.45	51.06	33.72	50.48
7	32.97	28.32	27.96	27.24	36.20	29.63	30.39
8	69.23	70.00	59.23	63.08	63.85	63.41	64.80
9	57.58	56.15	50.82	50.61	57.79	54.47	54.57
10	66.50	73.10	61.17	68.78	67.01	68.73	67.55
11	32.12	33.73	29.76	30.73	30.41	31.70	31.41
12	70.23	69.77	70.68	69.92	68.57	67.38	69.42
13	57.66	58.00	56.97	56.37	55.00	59.35	57.23
14	38.67	38.28	36.13	39.65	39.84	31.87	37.41
15	66.07	67.86	60.20	64.54	60.97	61.32	63.49
16	51.83	51.83	48.35	47.83	47.13	52.84	49.97
17	51.22	49.86	47.97	49.05	50.14	51.78	50.00
<b>Avg</b>	52.57	54.29	48.82	51.02	50.63	49.39	51.12

usage.

**Set #5** used a labeling scheme which consisted of mapping each sign to its vocabulary label and included out of vocabulary signs. The time segmentations (sign boundaries) were labeled automatically by HTK. This set serves as a reference point for some earlier experimentation methods.

**Set #6** used a labeling scheme which consisted of mapping each sign to its vocabulary label but does not include out of vocabulary signs. The time segmentations (sign boundaries) were labeled automatically by HTK. This set serves as a reference point for some earlier experimentation methods.

### 6.2.2 Results

Tables 16, 17, and 18 show the results of the leave-one-out by child experiments. One of the interesting things that one can see in this graph is the wide variation of performance by child. From Table 16, which shows word accuracy, the best performing child set is Child #4 with an average rate of 73.90% and a maximum rate of 78.70%. The lowest performer

**Table 17:** Percentage word correct for per student, per scheme.

Child	Set #0	Set #1	Set #2	Set #3	Set #5	Set #6	Avg
0	37.11	42.14	39.62	38.99	37.11	44.80	39.96
1	61.08	63.17	54.49	53.59	54.19	57.70	57.37
2	63.66	67.59	61.65	65.87	60.79	66.49	64.34
3	52.89	57.02	47.11	50.41	49.59	55.77	52.13
4	75.15	80.18	73.37	76.92	74.56	69.64	74.97
5	51.46	51.94	44.95	53.07	46.74	46.44	49.10
6	62.77	67.02	61.70	65.96	64.89	61.63	63.99
7	58.78	55.91	52.69	53.05	54.12	56.79	55.22
8	72.31	73.85	62.31	66.15	70.77	67.48	68.81
9	69.26	69.06	64.14	65.16	68.65	69.72	67.67
10	76.40	81.73	71.32	77.41	72.59	74.16	75.60
11	35.22	38.54	34.05	35.97	35.01	40.35	36.52
12	70.83	70.68	71.73	70.98	69.62	68.15	70.33
13	63.73	64.16	63.39	62.96	59.97	70.17	64.06
14	54.88	55.08	49.41	53.91	50.98	53.37	52.94
15	68.62	71.94	62.24	68.37	63.01	63.16	66.22
16	58.43	61.74	55.30	57.04	56.52	61.64	58.44
17	59.89	61.25	56.64	56.91	58.27	59.47	58.74
<b>Avg</b>	60.69	62.94	57.01	59.60	58.19	60.39	59.80

**Table 18:** Percentage sentence correct for per student, per scheme.

Child	Set #0	Set #1	Set #2	Set #3	Set #5	Set #6	Avg
0	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1	4.55	2.27	2.27	0.00	2.27	0.00	1.89
2	5.04	7.91	2.16	6.47	3.60	5.76	5.16
3	0.00	0.00	0.00	0.00	0.00	0.00	0.00
4	15.38	17.31	15.38	13.46	13.46	3.85	13.14
5	0.75	0.75	0.75	2.24	0.00	1.49	1.00
6	0.00	7.14	7.14	14.29	7.14	0.00	5.95
7	0.00	0.00	0.00	0.00	0.00	0.00	0.00
8	15.00	25.00	15.00	20.00	20.00	5.00	16.67
9	3.95	2.63	2.63	2.63	5.26	10.53	4.61
10	8.47	15.25	3.39	16.95	8.47	6.78	9.88
11	0.00	0.85	0.00	0.00	0.00	1.71	0.43
12	8.70	13.04	11.96	15.22	10.87	6.52	11.05
13	3.85	2.31	3.08	2.31	1.54	6.15	3.21
14	0.00	0.00	0.00	0.00	0.00	0.00	0.00
15	5.08	10.17	0.00	5.08	5.08	3.39	4.80
16	1.30	1.30	1.30	3.90	2.60	3.90	2.38
17	2.00	4.00	2.00	4.00	2.00	0.00	2.33
<b>Avg</b>	4.11	6.11	3.73	5.92	4.57	3.06	4.58

is Child #3 with an average rate of 28.16% and a minimum rate of 25.62%

The overall average performance of the children for word accuracy is 51.12%. There is a 3.27 point spread across the average word accuracy rates for the top three performers by set: Set #1 at 54.29%, Set #0 at 52.57%, and Set #3 at 51.02%. This difference between the sets is surprisingly small.

### 6.3 Testing on Training

As discussed in Chapter 5, the testing on training experiments allow us to use the entire data set for both testing and training. This configuration is often used to give a reasonable “best case” baseline for recognition rates.

#### 6.3.1 Results

Table 19 shows the results of the testing on training recognition experiments. Set #3 is the best performer for the three measurements for this experiment, though the difference in performance between Set #1 and Set #3 is less than 3 points for all three measurements. It is surprising that Set #3 would be the best, considering how many classes it has and, correspondingly, how many examples per class. The subdivision of the signs into classes that include both handedness and quality would logically make it harder to differentiate between each class, especially between examples that are labeled as GOOD versus OK. Yet, the results are comparable, if not better than, the other sets.

#### 6.3.2 Analysis of Chance

Table 20 shows the chance of randomly choosing the correct class label for any given sign for each set. The “easiest” chance guess is Set #0 with a 3.57% chance, and the “hardest” guess

**Table 19:** Testing on training results

	<b>Word Accuracy</b>	<b>Word Correctness</b>	<b>Sentence Correctness</b>
<b>Set #0</b>	63.57	68.98	8.82
<b>Set #1</b>	68.98	73.48	12.59
<b>Set #2</b>	64.00	69.28	9.66
<b>Set #3</b>	70.97	75.32	14.02
<b>Set #5</b>	60.78	64.67	7.39
<b>Set #6</b>	61.19	67.80	7.30

**Table 20:** Chance of randomly choosing the right class assignment for a sign

Set	Classes	Chance	% Chance
Set #0	28	1/28	3.57%
Set #1	51	1/51	1.96%
Set #2	70	1/70	1.43%
Set #3	120	1/120	0.83%
Set #5	28	1/28	3.57%
Set #6	21	1/21	4.76%

**Table 21:** Testing on training results: Comparison with previous methods. Effects of game only vocabulary (Set #6) and full sign label set (Set #5) on auto-segmented labels. Effects of auto-segmentation (Set #5) and manually determined sign boundaries (Set #0).

Comparison	$\Delta$ Word Accuracy	$\Delta$ Correctness Correctness	$\Delta$ Sentence Correctness
Set #5 - Set #6	-0.41	-3.13	0.09
Set #0 - Set #5	2.79	4.31	1.43

is Set #3 with a 0.83% chance. The math is a slight oversimplification of the problem, since our classification is done in the phrase context. When the recognition is done in context, the recognition algorithm must decide both how to segment the phrase and the label for each segment. A rough estimate of the phrase level difficulty could be considered  $P(label)^3$ ,  $P(label)^4$ , and  $P(label)^5$  for a three, four, and five sign phrase. This estimate is a severe lower bound, since we have 9055 labeled segments for 1191 phrases, which averages 7.60 segments per phrase.

The small margin between Set #3 and Set #1 can be considered quite a large gain, since Set #3 is a more difficult problem and provides more information about the sign example. This extra information about handedness and sign quality could be incorporated into the game system to provide more detailed feedback to educators or to set an adjustable quality threshold for acceptable signs as GOOD, OK, or BAD.

### 6.3.3 Comparison to Previous Work

Table 21 shows a comparison of some techniques from previous work. Set #5 and Set #6 are similar to previous techniques used by our automatic recognition group (including Telesign and CopyCat) [131, 17, 95, 18]. They have been included in this set as a baseline for comparison.

**Table 22:** Game grammar by Level: Words in brackets ( [ ] ) are considered optional. Level 1 requires a three-sign phrase, with both adjectives optional. Level 2 requires a four-sign phrase, with the first adjective optional. Level 3 requires a five-sign phrase, with all signs required.

Level	Grammar				
Level 1	[Adjective 1]	Subject	Preposition	[Adjective 2]	Object
Level 2	[Adjective 1]	Subject	Preposition	Adjective 2	Object
Level 3	Adjective 1	Subject	Preposition	Adjective 2	Object

Our previous works have used either a statistical or structured grammar during the recognition phase. These grammars were appropriate to the defined language tasks and structure in the experiments. The applications of grammars for recognition are a well-know method to improve results in data sets with well structured language. These grammars are customized to the task and do not usually generalize well to other data sets [68, 161]. Table 22 shows an example of a generalized part-of-speech grammar that has been used in previous experiments.

As discussed in Chapter 4 Section 4.7, I have chosen to use a simple unrestricted grammar during my recognition experiments:  $\langle CLASS \rangle$ . My hypothesis was that the resulting rates for Set #6 would be much lower than achieved in previous work without this grammar. The testing on training results show that Set #6 is the one of the bottom two performers across all three metrics. Set #6 has the second lowest word rates and the lowest sentence rate.

The difference between Set #6 and Set #5 is the vocabulary used. Set #6 contains strictly the game vocabulary without any of the added classes for disfluencies or out of game signs. Set #5 uses the full set of labels generated in Chapter 4. My hypothesis was that the added vocabulary would improve rates. The results were mixed. The change resulted in lower word rates, but a slightly higher sentence rate. The auto-segmentation procedure for HTK clearly had difficulty in segmenting the new classes, but did better on some phrases as a whole.

The difference between Set #5 and Set #0 is the time segmentation. Set #5 uses the auto-segmentation procedure from HTK in order to initialize and train the models. Set #6 uses the hand-labeled time labels. Auto-segmentation is used by many researchers with

**Table 23:** Testing on training results: Effects of added handedness label

Comparison	$\Delta$ Word	$\Delta$ Correctness	$\Delta$ Sentence
	Accuracy	Correctness	Correctness
Set #2 - Set #0	0.43	0.3	0.84
Set #3 - Set #1	1.99	1.84	1.43

**Table 24:** Testing on training results: Effects of added quality label

Comparison	$\Delta$ Word	$\Delta$ Correctness	$\Delta$ Sentence
	Accuracy	Correctness	Correctness
Set #1 - Set #0	5.41	4.5	3.77
Set #3 - Set #2	6.97	6.04	4.36

large data sets in order to reduce the amount of labor per sample [161]. Auto-segmentation only requires a sequential list of labels, which requires significantly less time than manually determining time boundaries for each sample.

#### 6.3.4 Analysis of Handedness

Table 23 shows a comparison between comparable sets with and without the handedness labels. The addition of handedness labeling did provide a small improvement for both sets. The larger improvement was Set #3 which used both handedness and quality.

#### 6.3.5 Analysis of Quality

Table 24 shows a comparison between comparable sets with and without the quality labels. Quality provided more substantial improvement in rates between sets. The improvement for Set #3 was again the best, but the improvement for Set #1 was only around 1.5 points less for the three rates.

### 6.4 Training Models with GOOD Samples

The original intent of differentiating sign quality was to increase our understanding of the data set content. The refinement of labeling offered the opportunity to test the effects of models that were trained on GOOD data. By choosing the samples labeled as GOOD, we simulate pruning out bad examples during the training process.

Previous research has shown that cleaning data can improve performance of classification algorithms [91, 102], and though cleaning data is frequently used in many machine learning

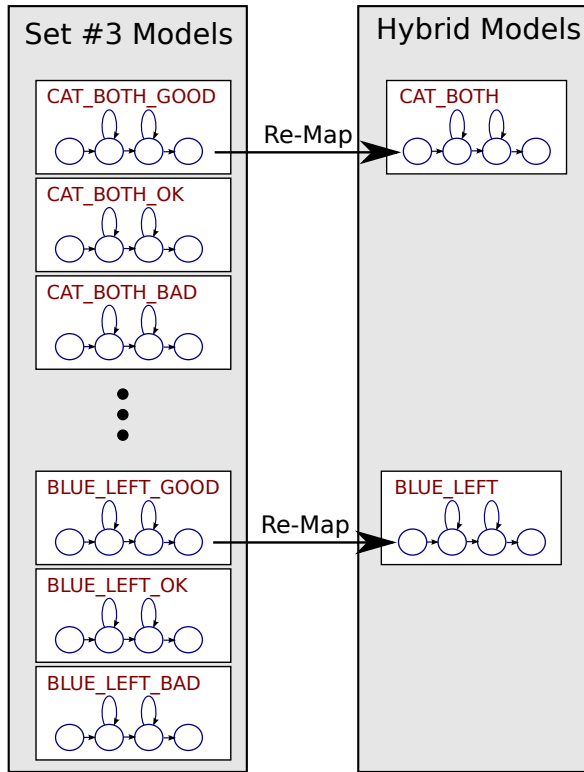
tasks, there is not a generalized rule for the process [8]. Most of the automatic sign language tasks defined in research are based on scripted, well-formed signing [88, 137], and some researchers are even using linguistic exemplar databases collected by sign linguists [99]. Signers in these data sets usually sign much slower than conversational signing by native signers [22]. These data sets are designed to provide good examples of signing, and bad examples are usually pruned out.

In contrast, our data set was collected from children playing a game. In Chapter 4 we found that 4% of our signs were considered of OK quality, and 2% were considered of BAD quality. In a data set with a non-trivial number of examples that were considered BAD or OK quality, discarding less well formed samples may substantially improve our ability to model the signs by reducing noise. The hypothesis was that these models would be more representative of good examples of the sign and might increase accuracy rates.

#### 6.4.1 Hybrid Experiments

Since handedness provided a clear benefit in the testing on training results, we chose to use Set #3 as the basis for the experiment. Set #3 provides us with classes that were differentiated by both handedness and quality. For each class from Set #1 (which had handedness), Set #3 would have three classes: a GOOD version, an OK version, and a BAD version.

Models were fully trained using the classes from Set #3. The models that represented GOOD classes were then re-mapped as the models for the entire class from Set #1 and then used to test the recognition. For example the Set #3 models would contain three models for left handed blue: BLUE\_LEFT\_GOOD, BLUE\_LEFT\_OK, and BLUE\_LEFT\_BAD. These Set #3 models can all be thought of as a division of the class for BLUE\_LEFT. The BLUE\_LEFT\_GOOD model would then be re-mapped to the entire BLUE\_LEFT class in Set #1. In this way models trained with the GOOD version from Set #3 would be used as the basis for testing the recognition. The result would be a hybrid experiment which would use the configuration for the labeling scheme of Set #1 for testing purposes, but use the GOOD models from Set #3 for the recognition. In this way we do a form of data cleaning,



**Figure 38:** Class re-mapping for the Hybrid experiments

by pruning out the OK quality and BAD quality examples during training. Figure 38 shows a diagram of the process.

#### 6.4.2 Results

The recognition results for the hybrid experiment were a word accuracy of 68.98%, word correctness of 73.84%, and sentence correctness of 13.10%. These rates are almost exactly those of Set #1, so much so that it is worth examining the numbers that generate those rates.

Experiment #1 testing on training output:

SENT: %Correct=12.59 [H=150, S=1041, N=1191]

WORD: %Corr=73.48, Acc=68.89 [H=6654, D=1450, S=951, I=416, N=9055]

Experiment hybrid testing on training output:

SENT: %Correct=13.10 [H=156, S=1035, N=1191]

WORD: %Corr=73.84, Acc=68.98 [H=6686, D=1387, S=982, I=440, N=9055]

The end result is that the number of hits (H), deletions (D), substitution errors (S), and insertion errors (I) differ so slightly that the rates are equivalent. Recognition on models trained of all qualities and those trained on only GOOD examples had the same performance. This similarity in results means that there is nothing to be gained in this approach with this data set, so I pursued other options. There are several possible reasons for the lack of improvement. Its possible that for a small enough vocabulary data pruning may not provide significant benefits in this case. The data set distribution has 94% GOOD samples, so the small number of OK and BAD examples may affect the representation of the models. Perhaps the inclusion of models for OK and BAD classes provides a better model for the variability in sign performance, resulting in a larger gain for differentiated models versus the data pruning approach.

## ***6.5 Recognition of Game Vocabulary***

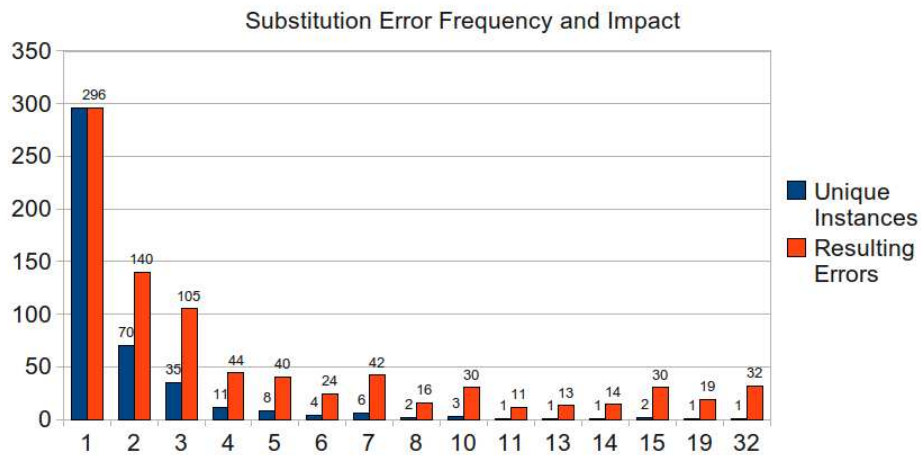
The confusion matrix for the testing on training recognition results can be used to give some insight into the recognition process. I will be looking at the results for the Experiment #3 in this section, since it is the most detailed labeling scheme and has the best performance. The matrix itself is not included in this section, since a class by class comparison for the experiment results in a confusion matrix of 120 rows by 120 columns which too large to display coherently. The overall counts for sign level recognition are Hits=6820, Deletions=1303, Substitutions=932, Insertions=394 for N=9055 sign examples across 120 sign classes.

### **6.5.1 Error Analysis: Substitution Errors**

Table 25 shows a listing of the number of substitution errors (a single entry in the confusion matrix) and the relative counts of those errors. Each instance of a substitution error (swapping label A for label B) is tallied by the confusion matrix. The table then shows the number of unique instances for each count of a substitution. There are 296 instances of specific substitution errors that occurred once (first row of Table 25) and one instance of a specific substitution error that occurred 32 times (last row of Table 25).

**Table 25:** A listing of the number of substitution errors (a single entry in the confusion matrix) and the relative counts of those errors. Each instance of a substitution error (swapping label A for label B) is tallied by the confusion matrix. The table shows the number of unique instances for each count of a substitution. There are 296 instances of specific substitution errors that occurred once (first row) and one instance of a specific substitution error that occurred 32 times (last row).

# of Errors	Instances	Resulting Errors
1	296	296
2	70	140
3	35	105
4	11	44
5	8	40
6	4	24
7	6	42
8	2	16
10	3	30
11	1	11
13	1	13
14	1	14
15	2	30
19	1	19
32	1	32



**Figure 39:** Comparison of the substitution errors and their frequencies. The chart displays the data shown in Table 25

**Table 26:** A listing of the top substitution errors

#of	Err	Correct label	Classified as
32		GREEN_RIGHT_GOOD	BLUE_RIGHT_GOOD
19		ALLIGATOR_BOTH_RIGHT_GOOD	ALLIGATOR_BOTH_RIGHT_OK
15		START_SENTENCE_RIGHT_GOOD	SILENCE_RIGHT_GOOD
15		SILENCE_RIGHT_GOOD	END_SENTENCE_RIGHT_GOOD
14		END_SENTENCE_RIGHT_GOOD	START_SENTENCE_RIGHT_GOOD
13		BLUE_LEFT_GOOD	GREEN_LEFT_GOOD
11		END_SENTENCE_RIGHT_GOOD	CHAIR_BOTH_LEFT_GOOD
10		SILENCE_RIGHT_GOOD	START_SENTENCE_RIGHT_GOOD
10		GREEN_RIGHT_GOOD	THE_RIGHT_GOOD
10		GARBAGE_RIGHT_GOOD	SILENCE_RIGHT_GOOD
8		START_SENTENCE_RIGHT_GOOD	END_SENTENCE_RIGHT_GOOD
8		GREEN_LEFT_GOOD	BLUE_LEFT_GOOD
7		SILENCE_RIGHT_GOOD	FIDGET_RIGHT_GOOD
7		ON_BOTH_RIGHT_GOOD	SPIDER_BOTH_RIGHT_GOOD
7		END_SENTENCE_RIGHT_GOOD	SILENCE_RIGHT_GOOD
7		END_SENTENCE_RIGHT_GOOD	FIDGET_RIGHT_GOOD
7		BLUE_RIGHT_GOOD	THE_RIGHT_GOOD
7		BLACK_RIGHT_GOOD	SNAKE_RIGHT_GOOD
6		SILENCE_RIGHT_GOOD	GARBAGE_RIGHT_GOOD
6		ORANGE_RIGHT_GOOD	SNAKE_RIGHT_GOOD
6		BLUE_RIGHT_GOOD	BLUE_RIGHT_OK
6		ALLIGATOR_BOTH_RIGHT_OK	ALLIGATOR_BOTH_RIGHT_GOOD

The top specific substitution errors are shown in Table 26. This listing shows the correct label and the incorrect substitution that was made for the sign, as well as the number of times this specific swap occurred. The prevalence of non-game signs such as silence and the start and stop sentence gestures indicates that it is worth examining the recognition effects of including these signs. There are also several confusions between GOOD and OK signs. The confusion between ALLIGATOR\_BOTH\_RIGHT\_GOOD and ALLIGATOR\_BOTH\_RIGHT\_OK occurs in both directions in the listing on Table 26 for a total of 25 errors. One question to investigate is whether this is a problem specific to the sign for alligator or is it indicative of a larger issue.

The recognizer appears to have great confusion between the signs BLUE and GREEN. Table 26 has four rows which represent the confusion between the two signs. Further analysis showed that the BLUE / GREEN confusion occurs five times for a total of 59 substitution errors. That means that the confusion between the two signs BLUE and GREEN represent

**Table 27:** A listing of the all substitution errors between BLUE and GREEN

# of Errors	Correct label	Classified as
1	BLUE_LEFT_GOOD	GREEN_LEFT_OK
5	GREEN_RIGHT_GOOD	BLUE_RIGHT_OK
8	GREEN_LEFT_GOOD	BLUE_LEFT_GOOD
13	BLUE_LEFT_GOOD	GREEN_LEFT_GOOD
32	GREEN_RIGHT_GOOD	BLUE_RIGHT_GOOD
59	<b>Total</b>	

6.33% of the substitution errors.

The signs BLUE and GREEN are minimal pairs, which differ in construction only by hand shape. They are the only minimal pairs in the language used during game play. Minimal pairs are discussed in more depth in Chapter 2 Section 2.1.7.

### 6.5.2 Error Analysis: Quality

How well is Set #3 differentiating between the quality of signs? How many of the errors reported for Set #3 are confusions between sign quality? In order to evaluate these questions the recognition results for Set #3 were re-calculated ignoring errors in quality classification. In other words, errors where a sign was classified correctly by sign label and handedness, but not quality would be considered correct.

Results for Set #3:

SENT: %Correct=14.02 [H=167, S=1024, N=1191]

WORD: %Corr=75.32, Acc=70.97 [H=6820, D=1303, S=932, I=394, N=9055]

Results for Set #3 when quality is ignored:

SENT: %Correct=14.86 [H=177, S=1014, N=1191]

WORD: %Corr=76.42, Acc=72.06 [H=6920, D=1305, S=830, I=395, N=9055]

The overall word accuracy improves from 70.97% to 72.06%, which is only a 1.09 point change. Note that the change in these results are predominately due to substitution errors. Substitution errors are reduced by 102, and deletion errors are increased by two. The number of correct hits increases from 6820 to 6890, for a net gain of 100 correct words. These results show that errors based on a misclassification of quality are a small minority of the recognition errors for Set #3.

### 6.5.3 Error Analysis: Handedness

How well is Set #3 differentiating between the handedness of signs? How many of the errors reported for Set #3 are confusions between sign handedness? In order to evaluate these questions the recognition results for Set #3 were re-calculated ignoring errors in handedness classification. Thus errors where a sign was classified correctly by sign label and quality, but not handedness would be considered correct.

Results for Set #3:

SENT: %Correct=14.02 [H=167, S=1024, N=1191]

WORD: %Corr=75.32, Acc=70.97 [H=6820, D=1303, S=932, I=394, N=9055]

Results for Set #3 when handedness is ignored:

SENT: %Correct=14.53 [H=173, S=1018, N=1191]

WORD: %Corr=75.55, Acc=71.15 [H=6841, D=1308, S=906, I=398, N=9055]

The overall word accuracy improves from 70.97% to 71.15%, which is only a 0.18 point change. Again the change in these results is predominately substitution errors. Substitution errors are reduced by 26, and deletion errors are increased by 5. The number of correct hits goes up from 6820 to 6841, for a net gain of 21 correct words. These results show that errors based on a misclassification of handedness are a very small minority of the recognition errors for Set #3.

### 6.5.4 Error Analysis: Non-sign gestures

The non-sign gestures in our labeling scheme are START\_SENTENCE, END\_SENTENCE, SILENCE, GARBAGE, FIDGET, and ERASE\_WAVE. These non-sign gestures account for 3157 signs in the data set, which is 34.68% of the labeled signs. Table 28 shows the error distribution across the non-sign gestures. Non-sign gestures accounted for 319 (34.22%) of the substitution errors, 667 (51.19%) of the deletion errors, and 132 (33.50%) of the insertion errors. Substitution errors between non-sign classes represented 124 (13.30%) of the substitution errors.

**Table 28:** Recognition errors for non-sign gestures. Substitution errors do not directly add to total because substitution errors between two non-sign gestures are double listed - once for each gesture.

Class	Substitution	Deletion	Insertion	% of Signs
START_SENTENCE	95	136	21	12.49%
END_SENTENCE	106	124	18	12.58%
GARBAGE	95	56	18	1.71%
SILENCE	144	326	35	6.70%
FIDGET	38	11	16	0.55%
ERASE_WAVE	34	14	24	0.83%
Non-sign Total	319	667	132	34.86%
All Total	932	1303	394	100%

Although non-sign errors do not account for an overly disproportionate number of recognition errors, they do affect the reported recognition rates. Further refinement of these classes could potentially improve our recognition rates and is introduced as future work in Chapter 8. The next question to examine is: how much do errors from non-sign gesture classes affect the mechanics of game play? We will re-examine our recognition experiments from this perspective in the next experiment.

## 6.6 Analysis of Class Variance

### 6.6.1 Class Structure

Each class in Set #1, Set#2, and Set #3 can be thought of as a subdivision of the classes in Set #0. In this way we can think of Set #0 as the parent set and the sets Set #1, Set#2, and Set #3 as the child sets. For example, the parent set contains a single class BLUE, but child Set #1 contains the corresponding classes {BLUE\_GOOD, BLUE\_OK, BLUE\_BAD}, child Set #2 contains the corresponding classes {BLUE\_LEFT, BLUE\_RIGHT}, and child Set #3 contains the corresponding classes {BLUE\_LEFT\_GOOD, BLUE\_LEFT\_OK, BLUE\_LEFT\_BAD, BLUE\_RIGHT\_GOOD, BLUE\_RIGHT\_OK, BLUE\_RIGHT\_BAD}.

One important question about the subdivision of the parent classes into child classes is whether we are creating better models. One way of estimating model improvement is to compare the variances of the observation probabilities for two models. The model with the smaller variances is modeling a narrower, more exact distribution and is probably modeling the data better.

**Table 29:** Comparison of observation variances that increase or decrease in Set #1, Set #2, and Set #3 with respect to the parent Set #0

Set	Increase	Decrease
Set #1	28.8%	71.2%
Set #2	20.0%	80.0%
Set #3	18.4%	81.6%

### 6.6.2 Class Variance

If we consider the definition of hidden Markov models in Chapter 2 Section 2.2.2, then we can examine the following:

- A set of observation probabilities  $P(v_k(t)|w_j(t)) = b_j(k)$  that represent the probability of making an observation  $v_k$  while in state  $w_j$  at time  $t$
- Let  $b_j(k)$  be a mixture of Gaussians defined by the means  $M^l = \{\mu_1, \mu_2, \dots, \mu_l\}$ , the variances  $\Sigma^l = \{\sigma_1^2, \sigma_2^2, \dots, \sigma_l^2\}$ , and the weights  $W^l = \{w_1, w_2, \dots, w_l\}$
- Let the total variance of the probability  $b_j(k)$  be the weighted sum  $x_j(k) = w_1\sigma_1^2 + w_2\sigma_2^2 + \dots + w_k\sigma_k^2$

We can do a state by state comparison of the total variance of the probabilities by each model. The scaling of each of the distributions  $b_j(k)$  will vary by  $k$ , since each  $b_j(k)$  represents the distribution in state  $j$  of feature  $k$ . This scaling difference prevents a simple averaging across  $k$ , but we can compare the vector  $b_j(k)$  for each model in the children Set #1, Set #2, and Set #3 to the vector  $b_j(k)$  for the corresponding classes in the parent Set #0. We can do this by using a simple ratio of  $CHILD[b_j(k)]/PARENT[b_j(k)]$  to test for an increase ( $> 1$ ) or decrease ( $< 1$ ) in the variance from the parent class, where  $CHILD[b_j(k)]/PARENT[b_j(k)]$  is the ratio of the weighted sums of the variances for model and state between the *CHILD* and *PARENT* models. Table 29 shows the aggregate results of these calculations across all models and all states for each set.

### 6.6.3 Results

Table 29 shows the percentage of the variances that increased or decreased for all features and all states, presented for each set as compared to Set #0. For all three child sets the variance for most of the observation features ( $> 70\%$ ) was reduced by the division of classes

from parent Set #0. Child Set #3 had the greatest reduction in variance with a decrease in 81.6% of the feature variances. Of the 126 features, 74 (58.7%) of them were reduced in all three sets, which indicates a good consistency in the feature modeling.

## ***6.7 Post-processing to Support Game Play***

### **6.7.1 Parsing for ASL Signs**

The recognition results reported thus far have been standard recognition results based on matching a set of classes to the output recognition transcripts. The confusion matrix analysis showed that non-sign gestures account for a large number of errors, even if they are relatively proportionate to their representation in the data set. This revelation inspires the question: How do the recognition results vary if I only consider game vocabulary? These non-sign gestures which show up in the literal transcriptions would be largely ignored as a function of phrase correctness while playing the game. A phrase that has fidgeting, silences, or extra mouse gestures would still be considered correct by a human.

To test for recognition results that more directly correspond to our intentions, for game play I created a parser which strips out the labeling artifacts from the recognition transcripts and reduces the classes to the sign vocabulary (including the out of vocabulary signs such as THE). These cleaned transcriptions can then be used for evaluation of the recognition in terms of the game vocabulary. Errors that are strictly related to non-sign gestures will then be ignored. These ignored errors include the relevant insertion errors, deletion errors, and substitutions which are between two non-sign gestures. Substitution errors for ASL signs to excluded classes will become deletion errors. Substitution errors for excluded classes to ASL signs will then become insertion errors.

### **6.7.2 Results**

Table 30 shows a comparison of the reporting methods: raw recognition results and the post-processing results which exclude non-sign gestures. The column for Set #6 stands out in this table for its very small differences between the raw recognition and the post-processing. Upon consideration this result makes sense because Set #6 didn't include any out of vocabulary signs. However, Set #6 did include the mouse gestures for start and stop

**Table 30:** Comparison of testing on training results with the raw recognition output and the post-processing output.

	Set #0	Set #1	Set #2	Set #3	Set #5	Set #6	Hybrid
<b>Word Accuracy</b>							
Raw	63.57	68.98	64.00	70.97	60.78	61.19	68.98
Post	68.64	74.26	70.50	77.22	65.06	61.57	74.49
$\Delta$	5.07	5.28	6.50	6.25	4.28	0.38	5.51
<b>Word Correctness</b>							
Raw	68.98	73.48	69.28	75.32	64.67	67.80	73.84
Post	72.93	79.18	75.45	82.07	68.41	68.12	79.27
$\Delta$	3.95	5.70	6.17	6.75	3.74	0.32	5.43
<b>Sentence Correctness</b>							
Raw	8.82	12.59	9.66	14.02	7.39	7.30	13.10
Post	13.60	20.40	14.86	23.85	10.83	7.39	21.24
$\Delta$	4.78	7.81	5.2	9.83	3.44	0.09	8.14

sentences, since these have been previously used as markers. The very small delta values can be explained by the few recognition errors that involved the start and stop gestures with the mouse. Additionally, Set #1 and the quality based hybrid experiment from Section 6.4 on page 103 remain very similar, even after the post-processing.

Set #3 remained the best performer and had the highest delta values for word accuracy and sentence correctness, with a very large 9.83 point jump in sentence correctness. Overall the boost in rates corresponded well to the confusion matrix information. Post-processing Set #3 results in a word accuracy of 77.22% which makes it the best candidate for the recognition scheme so far.

## 6.8 *N-Best Recognition*

N-best algorithms can yield higher raw accuracy rates and can increase the performance of systems that can evaluate multiple hypotheses. N-best filtering is frequently used in combination with natural language processing or other post-processing options to create more robust systems. Chapter 2 Section 2.2.3 contains a deeper discussion of the use of N-best filtering and re-scoring in speech and gestures systems.

The token passing algorithms used for recognition in HTK have the ability to pass multiple tokens and generate multiple N-best hypotheses. The word N-best algorithm used

**Table 31:** Percentage word accuracies for the N-best experiments

Set	N=1	N=2	N=3	N=4	N=5	N=6	N=7	N=8	N=9	N=10
#0	63.57	67.41	69.38	70.60	71.45	72.26	72.89	73.37	73.80	74.12
#1	68.89	72.41	74.20	75.45	76.17	76.89	77.47	77.97	78.38	78.83
#2	64.00	67.43	69.45	70.88	71.76	72.42	73.05	73.50	73.86	74.14
#3	70.96	74.63	76.22	77.34	78.46	79.07	79.59	80.06	80.45	<b>80.70</b>
#5	<b>60.78</b>	64.65	66.66	67.93	68.88	69.60	70.12	70.66	71.11	71.55
#6	61.19	65.65	68.10	69.43	70.70	71.62	72.35	72.93	73.38	73.84

**Table 32:** Percentage word correct for the N-best experiments

Set	N=1	N=2	N=3	N=4	N=5	N=6	N=7	N=8	N=9	N=10
#0	68.98	72.18	73.78	74.73	75.42	76.10	76.61	77.05	77.42	77.69
#1	73.48	76.36	77.85	78.90	79.48	80.12	80.57	80.99	81.30	81.67
#2	69.28	71.98	73.59	74.79	75.58	76.16	76.71	77.11	77.40	77.65
#3	75.31	78.34	79.67	80.65	81.59	82.08	82.51	82.93	83.26	<b>83.46</b>
#5	<b>64.67</b>	67.93	69.62	70.69	71.52	72.15	72.63	73.04	73.45	73.82
#6	67.80	71.52	73.48	74.45	75.47	76.20	76.83	77.35	77.74	78.17

in HTK has been shown to be empirically comparable in performance to an optimal N-best algorithm [161]. Can we use a combination of N-best multiple hypotheses with the construction of the automated game to compensate for recognition errors? First, we must evaluate the performance of the results of N-best recognition experiments on the data.

### 6.8.1 Experiments

In this section I will cover the recognition results for the  $N = 10$  experiments which were ultimately used to construct the automatic game (which is covered in Chapter 7). The experiments in this section are configured with two axes. The first axis is the variation of labeling scheme: Set #0, Set #1, Set #2, Set #3, Set #5, Set #6. These schemes are the same as presented in the previous sections. The second axis is the variation in the value of  $N$  during N-best, for  $N = 1$  through  $N = 10$ .

### 6.8.2 Results

Tables 31, 32, and 33 show the raw results of the recognition experiments. The maximum and minimums per table are both in bold. The minimums for all three tables are in Set #5 and Set #6, for  $N=1$ . Set #3 (highlighted) is the best performer across all three tables.

Tables 34, 35, and 36 show the results of the recognition experiments with post-processing.

**Table 33:** Percentage sentence correct for the N-best experiments.

Set	N=1	N=2	N=3	N=4	N=5	N=6	N=7	N=8	N=9	N=10
#0	8.82	13.52	16.04	17.21	18.30	18.98	19.82	20.74	21.66	21.91
#1	12.59	18.05	19.90	21.58	22.33	23.93	25.02	26.36	27.12	27.71
#2	9.66	13.18	15.62	17.38	18.56	19.31	19.98	20.49	21.16	21.83
#3	14.02	18.64	21.07	22.25	23.85	25.52	26.53	27.37	28.21	<b>28.88</b>
#5	7.39	10.92	12.59	13.18	14.19	14.95	15.37	16.20	16.54	16.88
#6	<b>7.30</b>	11.34	13.94	15.70	16.96	18.39	19.48	20.40	21.07	21.58

**Table 34:** Percentage word accuracy for the N-best experiments with post-processing.

Set	N=1	N=2	N=3	N=4	N=5	N=6	N=7	N=8	N=9	N=10
#0	68.64	72.58	74.62	75.89	76.74	77.51	78.13	78.70	79.23	79.60
#1	74.26	78.06	79.89	81.21	82.11	82.79	83.28	83.87	84.28	84.76
#2	70.50	73.95	76.02	77.58	78.52	79.20	79.71	80.05	80.46	80.73
#3	77.22	80.92	82.90	83.88	84.92	85.53	86.07	86.46	86.80	<b>87.10</b>
#5	65.06	69.32	71.51	72.93	74.08	74.86	75.45	75.97	76.41	76.90
#6	<b>61.57</b>	65.97	68.37	69.70	70.94	71.89	72.62	73.13	73.61	74.08

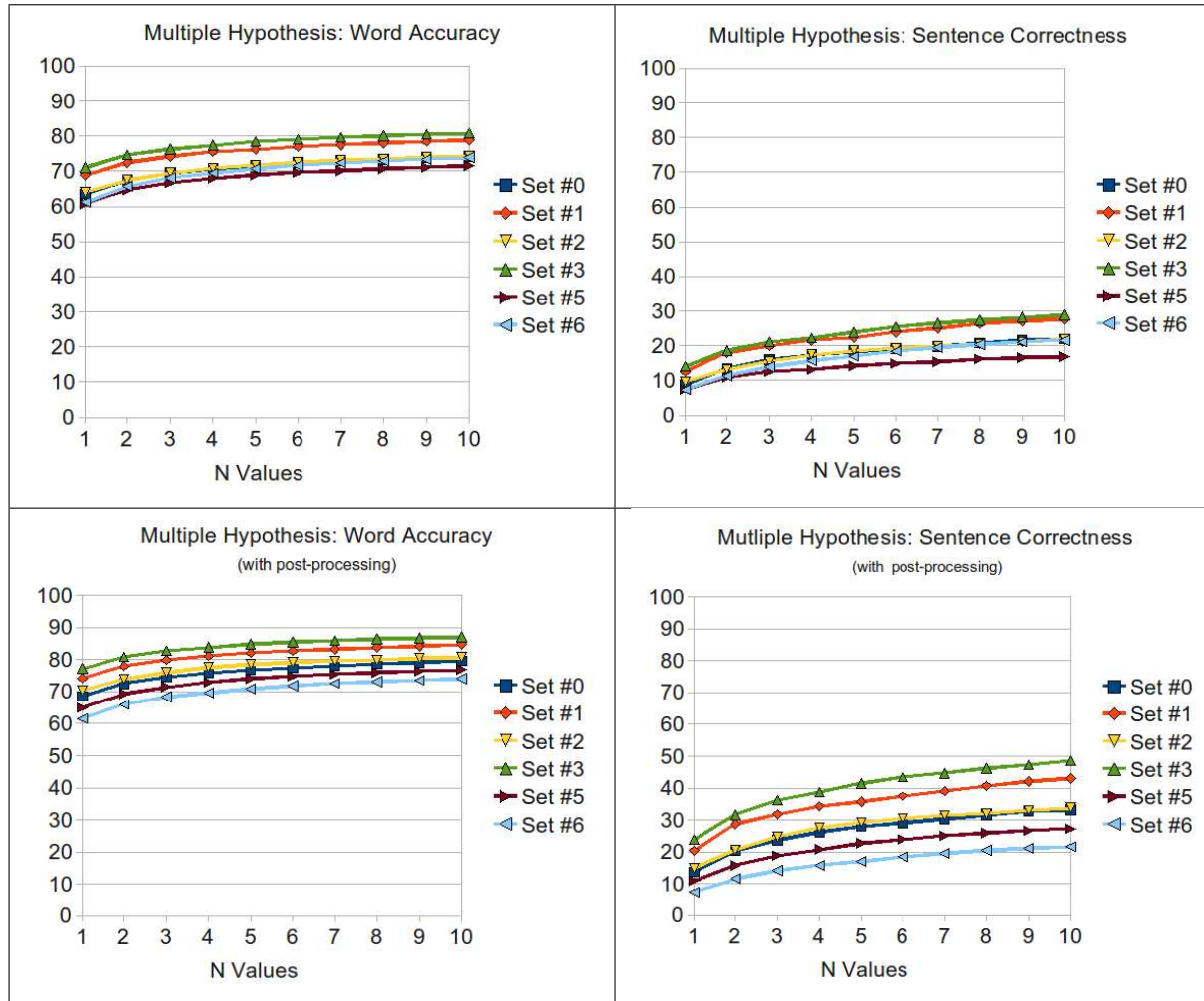
**Table 35:** Percentage word correct for the N-best experiments with post-processing.

Set	N=1	N=2	N=3	N=4	N=5	N=6	N=7	N=8	N=9	N=10
#0	72.93	76.21	77.81	78.82	79.57	80.20	80.72	81.22	81.68	81.99
#1	79.18	82.16	83.62	84.67	85.35	85.96	86.33	86.80	87.11	87.50
#2	75.45	78.20	79.77	81.10	81.89	82.46	82.91	83.16	83.48	83.69
#3	82.07	84.95	86.56	87.38	88.18	88.59	89.03	89.33	89.61	<b>89.85</b>
#5	68.41	72.01	73.89	75.13	76.08	76.74	77.25	77.64	78.03	78.50
#6	<b>68.12</b>	71.79	73.70	74.67	75.66	76.40	77.01	77.49	77.91	78.36

**Table 36:** Percentage sentence correct for the N-best experiments with post-processing.

Set	N=1	N=2	N=3	N=4	N=5	N=6	N=7	N=8	N=9	N=10
#0	13.60	20.40	23.68	26.20	27.96	29.05	30.31	31.57	32.66	33.17
#1	20.40	28.72	31.82	34.34	35.68	37.53	39.04	40.64	42.07	42.99
#2	14.86	20.57	24.77	27.54	29.22	30.56	31.40	31.99	33.00	33.75
#3	23.85	31.65	36.27	38.79	41.48	43.41	44.67	46.18	47.36	<b>48.53</b>
#5	10.83	15.79	18.81	20.74	22.67	23.85	25.02	25.94	26.62	27.29
#6	<b>7.39</b>	11.59	14.11	15.79	17.04	18.47	19.56	20.49	21.16	21.66

Set #3 (highlighted) is again the best performer across all tables, and Set #6 is the low performer.



**Figure 40:** Charts for N-best recognition rates. The first row shows the recognition rates before post-processing. The second row shows the rates after post-processing.

**Table 37:** Time in microseconds for recognition of a single phrase. Tests were run on a AMD Athlon™64 Processor 3200+ 1000MHz with 1 Gig of RAM

	<b>Average</b>	<b>Min</b>	<b>Max</b>
<b>N=1</b>	8.27	6	16
<b>N=2</b>	8.44	6	19
<b>N=3</b>	8.67	6	19
<b>N=4</b>	8.85	7	19
<b>N=5</b>	9.13	6	23
<b>N=6</b>	9.38	6	25
<b>N=7</b>	9.61	6	28
<b>N=8</b>	10.06	6	29
<b>N=9</b>	10.46	7	33
<b>N=10</b>	11.03	6	38

The maximum for  $N$  is set at 10 because as  $N$  grows, the amount of time required for recognition grows as well. Table 37 shows a chart of the times for running at  $N = 1$  through  $N = 10$ . These times were calculated by using the “time” program in Linux to calculate CPU time for a recognition call for a single sample. The time program was run for every sample in the set, so Table 37 represents aggregate statistics across the entire set of samples used in this chapter.

## **6.9 Summary**

The progression of experiments shows Set #3 to be a consistent high performer, which was surprising due to the large number of classes. The success of Set #3 indicates that not only can we train models to recognize the signs, but adding quality measurements improves the recognition rates. A post-processing step to eliminate errors from non-sign gestures showed that the word accuracy rates for Set #3 were increased 6.25 percentage points to 77.22%. N-best experiments resulted in even higher recognition rates, reaching a word accuracy of 87.10%. Labeling Set #3, with N-best lists for  $N = 10$ , as well as the parsing information from the post-processing experiments, will be integrated into the design of the automatic game in the next chapter.

## CHAPTER VII

### AUTOMATING GAME PLAY

#### *7.1 Introduction*

The work for this dissertation, on many levels, can be summarized as replacing the linguist who acted as the Wizard in our preliminary studies. The recognition work, combined with the work in this chapter, are ultimately designed to create a complete running CopyCat system for deployment and testing with live children. There has been a great amount of research on evaluating live system testing for dialogue systems that have been designed using Wizard of Oz systems [15, 66, 72, 114, 148]. There is a tension between domain specific criteria, intermediary evaluation and metrics, human judgement, and the input / output mapping of the final system. Boisen and Bates propose a framework that separates the implementation from the task and focuses on the mapping of reference criteria to the comparison program output [15]. For a deeper discussion of this related work see Chapter 2 Section 2.2.4 on page 21.

We have made several intermediary evaluations of the CopyCat system based on the following criteria: our system's educational effectiveness ([19, 149], see Chapter 3), the usability of the system ([58, 57, 80, 81], see Chapter 3) and automatic sign language recognition ([18, 163, 164], see Chapter 6). These intermediary evaluations are particularly important due to the rapid iterative development that resulted in the first two project deployments.

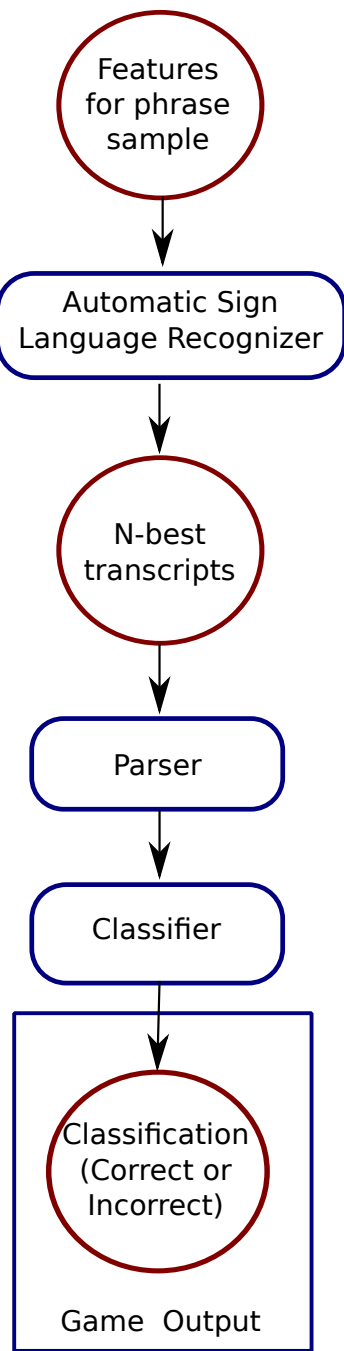
I propose that we use the framework of Boisen and Bates [15] to evaluate my work on replacing the Wizard. In this chapter, the comparison program can be considered the automatic game. I will use two versions of reference criteria for evaluation. First, I will use the human classification of the phrases collected during the second deployment in order to understand how well the system works in a strict language correctness model. If the transcription produced by the automatic game reflects a valid ASL representation of each game encounter, then it will be accepted as correct by the system; otherwise it will be

classified as wrong. Second, I will use the Wizard classification as a reference criteria in order to understand how the automatic game compares to the live Wizard. I will examine the mapping of the comparison program classification to the reference criteria and profile the performance by correct phrase categorizations, false positives, false negatives, and accuracy.

## 7.2 *Research Goals*

The objectives of this chapter can be summarized

- Contributions:
  - I will use the ontology to design a representative and flexible language processing component for the CopyCat game.
- Research Questions:
  - Can our data characterization help to create a language parser that is representative of language use in our data set?
  - How do the previous steps change the CopyCat game?
- Hypotheses:
  - Improved labeling schemes will help bring us closer to mimicking the performance of a live Wizard.
  - The data characterization can be used to create a representative parser.
- Methods:
  - Design a parser and classifier which complete the automatic game by outputting a classification of correct or incorrect
  - Evaluation of automatic game performance using hand-labeled ground truth as input
  - Evaluation of automatic game performance using the recognition engine results as input
  - Profile of Wizard performance compared with hand-labeled ground truth
- Data Collected:



**Figure 41:** Constructing the automatic game

- Metrics will include accuracy rates as well as true positive, false negative, true negative, and false positive
- Analysis:
  - Structure and function of parser and classifier components of automatic game
  - Comparison of accuracy rates as well as true positive, false negative, true negative, and false positive rates for the live Wizard and the automatic game

### 7.3 *Characterizing the Human Wizard*

#### 7.3.1 Wizard Behavior

Our language assessment showed an improvement in the children’s language skill after playing CopyCat with a Wizard [149, 19]. Thus, we will use the Wizard’s performance profile as a standard to which we hope to attain. We can then use the Wizard’s performance profile as a standard which has a known positive effect. I plan on profiling the Wizard’s decisions and comparing them to the recognition results and retrospective evaluations of the children’s signing. Since the baseline for system performance is a comparison to the Wizard’s performance, we must evaluate the Wizard’s activities. This comparison will involve an analysis of the Wizard’s response to each phrase.

In order to replace the Wizard, I needed to articulate what the criteria were for accepting or rejecting a sample. This process was conducted through email exchanges and face to face meetings. The Wizard discussed his criteria, and I attempted to deconstruct it into a series of rules. The Wizard would then review the rules and provide feedback for further refinement.

The Wizard accepted

- clear and understandable sign
- repetitions of correct words

**Table 38:** Profile of the Wizard’s performance

	<b>True</b>	<b>False</b>
<b>Positive</b>	1134	57
<b>Negative</b>	307	54

- correct signs preceded by incorrect signs of the same group (verb, adjective, noun, preposition)
- resets from an erase wave
- reset any time during the sentence (if last part of sentence is correct, previous signs do not matter)

The Wizard rejected

- series of signs from the same group that end in an incorrect sign
- extremely poor sign formation of multiple signs
- multiple self corrections on higher levels

This criteria for accepting or rejecting a phrase was then used for doing a ground truth labeling of the data set. Each phrase was reviewed and labeled as correct or incorrect. This retrospective review had the benefits of instant replay, slow motion step through, and frequent breaks. In contrast, the Wizard classified each example as it happened, with no opportunity to replay and with breaks only between gaming sessions.

### 7.3.2 Evaluation Metrics

The Wizard's performance was compared to the ground truth using the following phrase-level metrics

- Let  $P$  be true positives,  $N$  be all true negatives,  $TP$  be true positive,  $FP$  be false positive,  $TN$  be true negative, and  $FN$  be false negatives
- **Accuracy** =  $(TP + TN)/(P + N)$
- **True Positive Rate (Sensitivity or Recall)** =  $TP/P = TP/(TP + FN)$
- **False Negative Rate** =  $FN/P = FN/(TP + FN)$
- **False Positive Rate (Fall out)** =  $FP/N = FP/(FP + TN)$
- **True Negative Rate (Specificity)** =  $TN/N = TN/(FP + TN)$

### 7.3.3 Wizard Performance

The ground truth labeling showed that we had 1188 correct examples and 364 incorrect examples. Therefore the data set had an approximately 3:1 ratio of correct to incorrect

samples. Table 38 shows the profile of the Wizard’s performance. The Wizard had an overall accuracy of 92.85%. This performance was very good considering how demanding the Wizard’s job was. The Wizard’s performance metrics are then calculated as:

- Let  $P$  be true positives,  $N$  be all true negatives,  $TP$  be true positive,  $FP$  be false positive,  $TN$  be true negative, and  $FN$  be false negatives
- Accuracy = 92.85%
- True Positive Rate (Sensitivity or Recall) = 95.45%
- False Negative Rate = 4.55%
- False Positive Rate (Fall out) = 15.66%
- True Negative Rate (Specificity) = 84.34 %

#### **7.3.4 Bias of the Space**

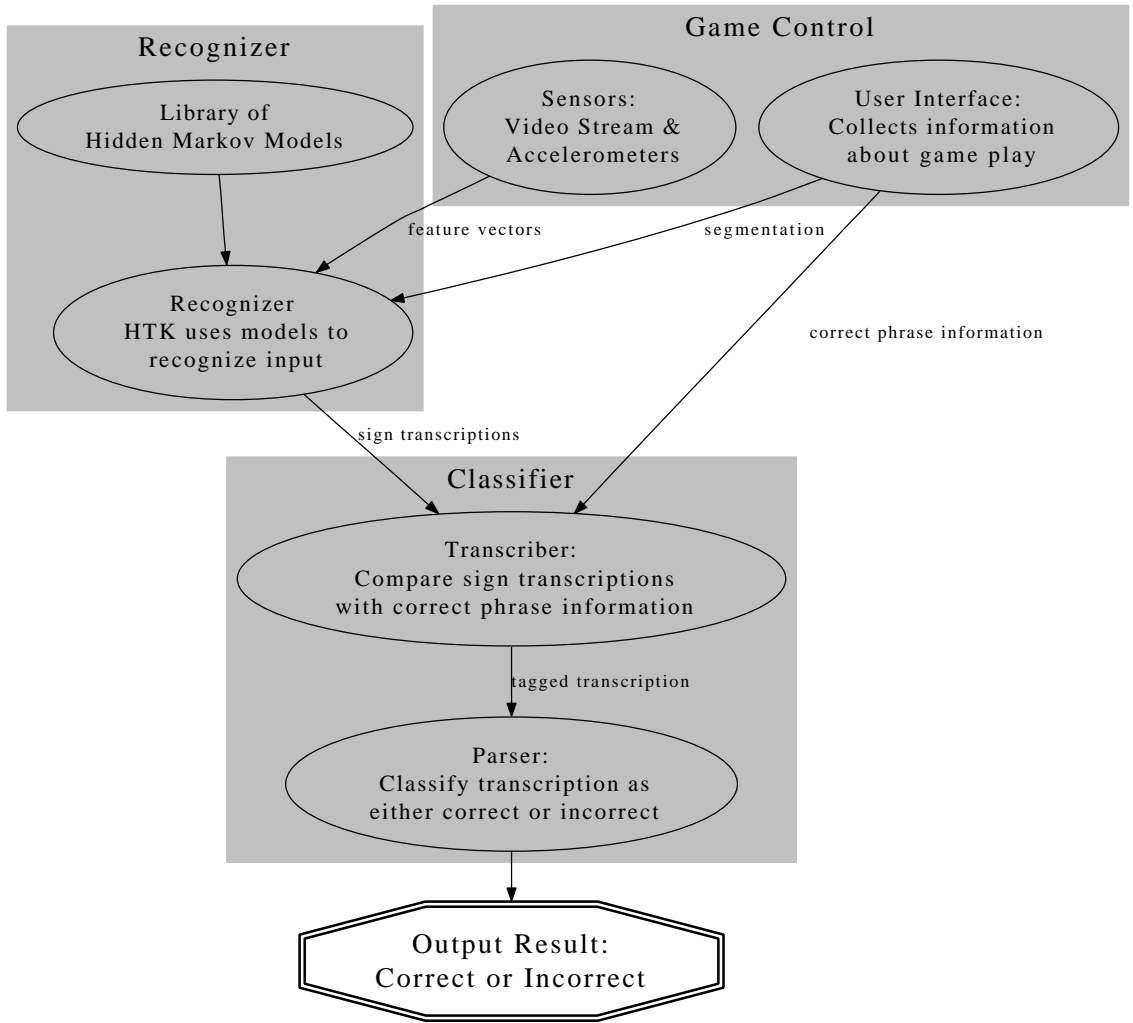
It is important to characterize the bias of the data space to provide context for further analysis. By classifying every answer as correct we can test the positive bias of the space. The result is a 76.53% accuracy, 100% true positive rate, and a 100% false positive rate. This test demonstrates a fairly strong positive bias for the data set. Errors on positive examples (false negatives) will have a disproportionate impact on system accuracy. Errors on negative examples (false positives) will have a much more minimal impact on system accuracy. The ratio of positive to negative examples is approximately 3:1.

### **7.4 Automating Game Responses**

#### **7.4.1 Architecture**

Figure 42 shows a high-level structural view of the entire system assembled. The high-level *Game Control* was discussed in Chapter 3, with an in-depth examination of the *Sensors* in Chapter 5. Development and testing of the *Recognizer* was discussed in the previous chapter, Chapter 6. Now we examine the Classifier and the Parser, which complete the system’s ability to mimic the Wizard’s behavior.

Aist and Mostow describe design approaches for educational spoken dialogue systems and include a categorization of children’s answers to questions in prototyped dialogue systems [4, 5]. The categories are:



**Figure 42:** Diagram of data flow for automating the game

- **Predicted=**: exact match
- **Predicted $\sim$** : a match with a morphological variation
- **Predicted+**: a match with additional material
- **Correct**: correct but not predicted
- **Incorrect**: incorrect

These categories provide an interesting point of reference for the CopyCat project. Our previous automatic game uses phrase verification to classify answers as correct or incorrect [163, 164]. In this approach, the automatic game functions more as a sentence repetition task than a dialogue system since it accepts answers that would only be classified as

“Predicted=”. My goal in designing the Classifier and Parser is to extend the language acceptance model to include “Predicted=”, “Predicted~”, and some “Predicted+”, while minimizing the set of “Correct” (but not predicted).

#### 7.4.2 Design

In order to replace the verification system with automatic sign recognition, the architecture of the game must change. I designed a parser and classifier to replace the verification components of the system. First, I characterized the language usage within the data set through interviews with the Wizard about his game criteria. These interviews helped me create a set of rules for accepting or rejecting a sample.

I used these rules to design a state machine for language parsing. The state machine

- Accepts samples that are exactly correct

*Ex. SPIDER IN GREEN BOX*

- Accepts samples that follow ASL grammar and are semantically correct

*Ex. SPIDER IN BOX GREEN*

- Accepts self corrections at sign or clause level

*Ex. ALLIGATOR SILENCE SPIDER IN GREEN BOX*

*Ex. ALLIGATOR IN SILENCE SPIDER IN GREEN BOX*

- Disregards disfluencies in most cases

*Ex. FIDGET SPIDER SILENCE IN BOX*

- Accepts most variations in sign

The Wizard also made some decisions using an element of human judgment that is not represented in the state machine. Some phrases were rejected based on poor performance on higher levels, too many self-corrections, or too many poorly formed signs.

#### 7.4.3 Classifier

The Classifier sub-system compares the transcript output from the Recognition system and the correct transcription for the intended phrase. The Classifier will then tag correct

**Table 39:** Game vocabulary

Subject	Object	Adjective	Verb
ALLIGATOR	BED	BLACK	BEHIND
CAT	BOX	BLUE	IN
SNAKE	CHAIR	GREEN	ON
SPIDER	FLOWERS	ORANGE	UNDER
	WAGON	WHITE	
	WALL		

**Table 40:** Game grammar by Level: Words in brackets ( [ ] ) are considered optional. Level 1 requires a three-sign phrase, with both adjectives optional. Level 2 requires a four-sign phrase, with the first adjective optional. Level 3 requires a five-sign phrase, with all signs required.

Level	Grammar
Level 1	[Adjective 1] Subject Preposition [Adjective 2] Object
Level 2	[Adjective 1] Subject Preposition Adjective 2 Object
Level 3	Adjective 1 Subject Preposition Adjective 2 Object

matches for any vocabulary for the correct phrase. Table 39 shows a quick review of the game vocabulary.

Table 41: Examples of the Classifier’s tagging output. The transcripts are mapped to the correct phrase and correct matches are tagged by their group.

Correct Phrase	Transcription	Tagged Output	Notes
[GREEN] ALLIGATOR ON [BLUE] WALL	ALLIGATOR ON WALL	SUBJ PREP OBJ	Exact match
[GREEN] ALLIGATOR ON [BLUE] WALL	SPIDER ON WALL	SPIDER PREP OBJ	Signed SPIDER instead of ALLIGATOR
[GREEN] ALLIGATOR ON [BLUE] WALL	SPIDER ALLIGATOR ON WALL	SPIDER SUBJ PREP OBJ	Signed SPIDER and self corrected to ALLIGATOR
[GREEN] ALLIGATOR ON [BLUE] WALL	GREEN ALLIGATOR ON WALL	ADJ1 SUBJ PREP OBJ	Exact match
[GREEN] ALLIGATOR ON [BLUE] WALL	ORANGE ALLIGATOR ON WALL	ORANGE SUBJ PREP OBJ	Signed wrong color (ORANGE)
GREEN ALLIGATOR ON BLUE WALL	GREEN ALLIGATOR ON BLUE WALL	ADJ1 SUBJ PREP ADJ2 OBJ	Exact match

The game grammar (shown in Table 40) is used as a basis for accepting sentences in the Parser. Each grammar component (adjective 1, subject, preposition, adjective 2, object) can be thought of as a variable in the equation that defines the sentence. These variables are assigned different values for each phrase in the game.

When we refer to the correct phrase transcription, we can then use that content to assign values to the variables. The Classifier searches correct matches for these values and tags them by replacing them with their variable name. These tags represent signs that correctly match the signs from the correct phrase. Signs from the transcription that don't match the correct phrase get left in place. The resulting transcription has all the correct signs mapped to a grammar variable and the incorrect signs left in place. Table 41 shows several examples of tagged output examples. This tagged output is then passed on to the Parser to check against the rules for accepting the sentence as correct or rejecting it as incorrect.

#### **7.4.4 Parser**

The Parser examines the tagged output from the Classifier and checks it against its rules for accepting (classifying as correct) or rejecting (classifying as incorrect). The Parser's rules for accepting are based on the Wizard's criteria for accepting a phrase. These rules have been incorporated into a state machine (shown in Figure 43). The sentences are parsed backwards, starting with the last entry.

The state machine acts to build the sentence from the end moving forward. This approach emulates the Wizard's ability to accept self corrections on multiple grammatical levels, such as a single sign (GREEN BLUE), a clause (GREEN ALLIGATOR BLUE ALLIGATOR), or a sentence level (GREEN ALLIGATOR ON GREEN WALL ERASE\_WAVE BLUE ALLIGATOR ON GREEN WALL).

The Parser evaluates each entry and acts according to a further set of rules. If the entry is tagged, then the Parser passes it to the state machine. If the entry isn't tagged, the Parser performs conflict checking. The conflict checking ignores non-game vocabulary such as silence or fidgets and checks the game vocabulary words against the state to look for errors.

**Table 42:** Evaluation of automatic game performance on the hand-labeled transcripts

	<b>True</b>	<b>False</b>
<b>Positive</b>	1112	15
<b>Negative</b>	43	21

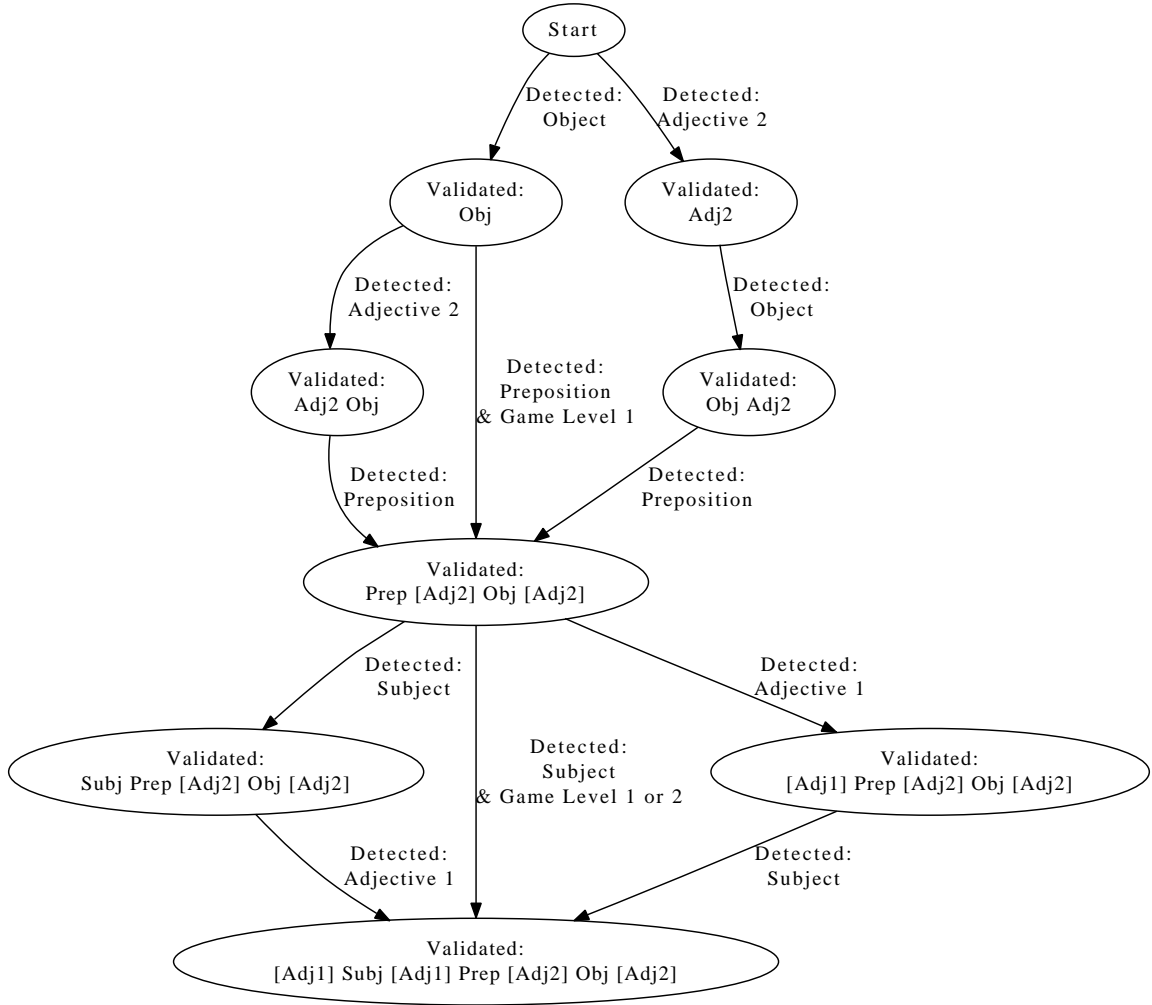
For example, if we are in the state where “Adjective 2” is validated, any preceding adjective is ignored (as a self-correction). In this state, the next valid option is the subject. If the entry hasn’t been tagged as the correct subject, then it is checked against the list of other subjects. If the entry matches an alternate subject, then the Parser will reject it as incorrect. Otherwise the Parser moves to the next entry.

As the Parser moves through the transcription, the state machine will build a memory of correct signs. At any entry the Parser could reject the sample due to a failed conflict check. If the Parser reaches the end of the transcription, it does a state check. If the sentence has been fully validated, then the Parser accepts the transcription. If the state machine is in some other state, then the Parser rejects the transcription.

#### 7.4.5 Baseline Performance

To establish a baseline for the performance of the Parser and Classifier systems, I used the hand-labeled transcripts as input to the Classifier. The results were compared to the hand-labeled ground truth. Using the raw count numbers shown in Table 42, the baseline performance metrics are then calculated by:

- Let  $P$  be true positives,  $N$  be all true negatives,  $TP$  be true positive,  $FP$  be false positive,  $TN$  be true negative, and  $FN$  be false negatives
- Accuracy = 96.98%
- True Positive Rate (Sensitivity or Recall) = 98.15%
- False Negative Rate = 1.85%
- False Positive Rate (Fall out) = 25.86 %
- True Negative Rate (Specificity) = 74.14%



**Figure 43:** State machine component of the Parser system. The state machine validates the transcription moving backwards through each entry.

The overall accuracy is 96.98%, which means that the automatic game is doing a very good job of evaluating the transcripts generated by the recognition step. Further examination of the discrepancies between the automatic game output and the hand-labeled ground truth showed that the errors were made on samples where human judgment of quality was the deciding factor. This observation means that the transcript may (or may not) have matched the required game language, but the Wizard rejected (or accepted) the sample based on a judgement call.

Overall, this kind of nuanced judgment affected  $FP + FN = 15 + 21 = 36$  samples which is 3.02% of the samples. Referring to the design approach for educational dialogue systems

by Aist and Mostow described in Section 7.4.1, we can see that the Parser and Classifier have done an excellent job of minimizing the set of transcripts that would be “Correct” (but not predicted).

## **7.5 Evaluation of the Automated Game**

The automated game system is shown in Figure 43. The process for testing the automated game using our archived data is as follows:

- Each sample was hand labeled (Chapter 4)
- Feature vector sets were generated for each sample (Chapter 5)
- The samples were used to build models for recognition (Chapter 6)
- The Recognizer was then used to generate transcripts of each sample (Chapter 6)
- These transcripts were then passed to the Classifier which tagged them and sent them to the Parser which returned a verdict of correct or incorrect. (Current Chapter)

### **7.5.1 Evaluation Metrics**

The procedure for evaluating the automated game is the same as the procedure for evaluating the Wizard discussed in Subsection 7.3.2.

### **7.5.2 N-Best Comparison**

N-best recognition was discussed in Chapter 6 Section 6.8. We used N-best recognition, where  $N = 10$  to generate the 10 optimal transcriptions for each sample. Each of those transcriptions was then passed through the Classifier and Parser. The results were evaluated by testing for  $N = 1$  through  $N = 10$  and *OR*-ing the results together. So for  $N = 1$ , the top transcription was used for evaluation. For  $N = 3$ , the top three transcriptions were compared. If any of the three transcriptions were classified as correct, the sample was decided correct for evaluation.

### **7.5.3 Automated Game Performance**

For the initial evaluation of the automated game performance I used the ground truth phrase labels as the reference criteria. This comparison shows how the automatic game performs

**Table 43:** Percentage accuracies for evaluation of automatic game accuracy. High and low values are in bold

Set	N=1	N=2	N=3	N=4	N=5	N=6	N=7	N=8	N=9	N=10
#0	42.55	48.48	51.45	53.38	55.51	56.74	57.70	58.87	59.64	60.41
#1	54.29	60.67	64.47	66.34	68.28	69.63	70.34	71.18	71.63	72.60
#2	44.62	49.32	53.00	55.32	57.45	58.28	59.57	60.28	61.19	61.64
#3	55.83	62.67	67.25	68.99	70.66	72.02	73.05	74.21	75.05	<b>75.63</b>
#5	38.88	44.10	47.90	49.45	51.13	52.61	53.38	54.42	55.13	55.90
#6	<b>35.65</b>	40.81	43.52	45.07	46.94	48.61	49.52	50.23	51.32	52.16

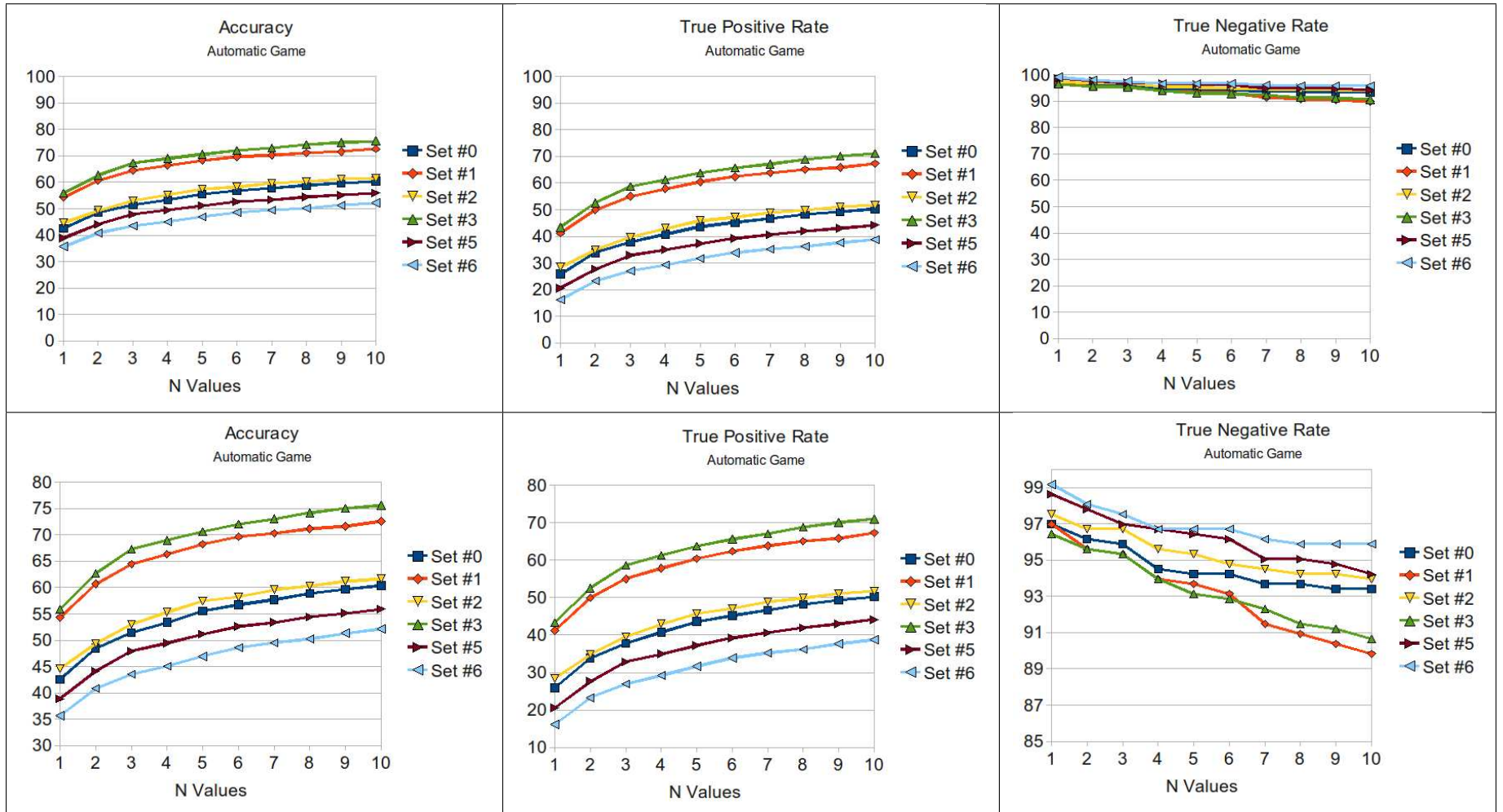
in the perspective of correct ASL language production. Tables 44, 43, and 45 show the results from evaluating the automated game with N-best recognition (training on testing set). Figure 44 contains charts which show the trends for accuracy, true positive rates, and true negative rates.

**Table 44:** Raw phrase counts for evaluation of automatic game evaluation.

	N=1	N=2	N=3	N=4	N=5	N=6	N=7	N=8	N=9	N=10
<b>Set #0</b>										
True +	307	402	449	484	518	537	554	572	585	597
False -	880	785	738	703	669	650	633	615	602	590
True -	353	350	349	344	343	343	341	341	340	340
False +	11	14	15	20	21	21	23	23	24	24
<b>Set #1</b>										
True +	489	593	653	687	718	741	758	773	782	799
False -	698	594	534	500	469	446	429	414	405	388
True -	353	348	347	342	341	339	333	331	329	327
False +	11	16	17	22	23	25	31	33	35	37
<b>Set #2</b>										
True +	337	413	470	510	544	559	580	592	606	614
False -	850	774	717	677	643	628	607	595	581	573
True -	355	352	352	348	347	345	344	343	343	342
False +	9	12	12	16	17	19	20	21	21	22
<b>Set #3</b>										
True +	515	624	696	728	757	779	797	818	832	843
False -	672	563	491	459	430	408	390	369	355	344
True -	351	348	347	342	339	338	336	333	332	330
False +	13	16	17	22	25	26	28	31	32	34
<b>Set #5</b>										
True +	244	328	390	415	442	466	482	498	510	524
False -	943	859	797	772	745	721	705	689	677	663
True -	359	356	353	352	351	350	346	346	345	343
False +	5	8	11	12	13	14	18	18	19	21
<b>Set #6</b>										
True +	192	276	320	347	376	402	418	430	447	460
False -	995	911	867	840	811	785	769	757	740	727
True -	361	357	355	352	352	352	350	349	349	349
False +	3	7	9	12	12	12	14	15	15	15

**Table 45:** Hit rates for evaluation of automatic game evaluation

	N=1	N=2	N=3	N=4	N=5	N=6	N=7	N=8	N=9	N=10
<b>Set #0</b>										
True +	25.86	33.87	37.83	40.78	43.64	45.24	46.67	48.19	49.28	50.29
False -	74.14	66.13	62.17	59.22	56.36	54.76	53.33	51.81	50.72	49.71
True -	96.98	96.15	95.88	94.51	94.23	94.23	93.68	93.68	93.41	93.41
False +	3.02	3.85	4.12	5.49	5.77	5.77	6.32	6.32	6.59	6.59
<b>Set #1</b>										
True +	41.2	49.96	55.01	57.88	60.49	62.43	63.86	65.12	65.88	67.31
False -	58.8	50.04	44.99	42.12	39.51	37.57	36.14	34.88	34.12	32.69
True -	96.98	95.6	95.33	93.96	93.68	93.13	91.48	90.93	90.38	89.84
False +	3.02	4.4	4.67	6.04	6.32	6.87	8.52	9.07	9.62	10.16
<b>Set #2</b>										
True +	28.39	34.79	39.6	42.97	45.83	47.09	48.86	49.87	51.05	51.73
False -	71.61	65.21	60.4	57.03	54.17	52.91	51.14	50.13	48.95	48.27
True -	97.53	96.7	96.7	95.6	95.33	94.78	94.51	94.23	94.23	93.96
False +	2.47	3.3	3.3	4.4	4.67	5.22	5.49	5.77	5.77	6.04
<b>Set #3</b>										
True +	43.39	52.57	58.64	61.33	63.77	65.63	67.14	68.91	70.09	71.02
False -	56.61	47.43	41.36	38.67	36.23	34.37	32.86	31.09	29.91	28.98
True -	96.43	95.6	95.33	93.96	93.13	92.86	92.31	91.48	91.21	90.66
False +	3.57	4.4	4.67	6.04	6.87	7.14	7.69	8.52	8.79	9.34
<b>Set #5</b>										
True +	20.56	27.63	32.86	34.96	37.24	39.26	40.61	41.95	42.97	44.14
False -	79.44	72.37	67.14	65.04	62.76	60.74	59.39	58.05	57.03	55.86
True -	98.63	97.8	96.98	96.7	96.43	96.15	95.05	95.05	94.78	94.23
False +	1.37	2.2	3.02	3.3	3.57	3.85	4.95	4.95	5.22	5.77
<b>Set #6</b>										
True +	16.18	23.25	26.96	29.23	31.68	33.87	35.21	36.23	37.66	38.75
False -	83.82	76.75	73.04	70.77	68.32	66.13	64.79	63.77	62.34	61.25
True -	99.18	98.08	97.53	96.7	96.7	96.7	96.15	95.88	95.88	95.88
False +	0.82	1.92	2.47	3.3	3.3	3.3	3.85	4.12	4.12	4.12



**Figure 44:** Charts for the Automated Game Accuracy Rates, True Positive Rates, and True Negative Rates testing. The first row shows the same Y-axis scale for all graphs. The second row shows zoomed versions with local scaling for the Y-axis.

**Table 46:** Percentage accuracies for comparison of automatic game to Wizard. High and low values are in bold

Set	N=1	N=2	N=3	N=4	N=5	N=6	N=7	N=8	N=9	N=10
#0	41.39	47.45	50.03	51.97	53.97	55.06	56.03	57.06	57.7	58.48
#1	53.26	59.51	62.93	64.8	66.47	67.44	68.28	69.25	69.7	70.66
#2	43.84	48.03	51.06	53.51	55.51	56.22	57.38	58.22	58.99	59.45
#3	54.8	61.64	65.57	67.44	68.86	70.08	71.12	72.15	72.73	<b>73.44</b>
#5	38.1	43.2	46.62	48.16	49.84	50.81	51.84	52.61	53.32	53.97
#6	<b>34.75</b>	39.39	41.72	43.26	45	46.55	47.58	48.03	49	49.84

#### 7.5.4 Comparison to Wizard

For the second evaluation of the automated game performance I use the results of the Wizard's live phrase classification during game play. This comparison shows how the automatic game performs relative to the live Wizard. This comparison is useful because the evaluations of educational effect have been performed for students playing the live game with the Wizard as the Classifier. Tables 47, 46, and 48 show the results from evaluating the automated game.

**Table 47:** Raw phrase counts for comparison of automatic game evaluation to Wizard.

	N=1	N=2	N=3	N=4	N=5	N=6	N=7	N=8	N=9	N=10
<b>Set 0</b>										
True +	300	396	440	475	508	526	543	560	572	584
False -	891	795	751	716	683	665	648	631	619	607
True -	342	340	336	331	329	328	326	325	323	323
False +	18	20	24	29	31	32	34	35	37	37
<b>Set 1</b>										
True +	483	586	643	677	706	726	744	760	769	786
False -	708	605	548	514	485	465	447	431	422	405
True -	343	337	333	328	325	320	315	314	312	310
False +	17	23	27	32	35	40	45	46	48	50
<b>Set 2</b>										
True +	333	405	457	498	531	545	565	578	591	599
False -	858	786	734	693	660	646	626	613	600	592
True -	347	340	335	332	330	327	325	325	324	323
False +	13	20	25	28	30	33	35	35	36	37
<b>Set 3</b>										
True +	509	618	685	718	745	766	784	804	816	828
False -	682	573	506	473	446	425	407	387	375	363
True -	341	338	332	328	323	321	319	315	312	311
False +	19	22	28	32	37	39	41	45	48	49
<b>Set 5</b>										
True +	240	323	382	407	434	454	472	486	498	511
False -	951	868	809	784	757	737	719	705	693	680
True -	351	347	341	340	339	334	332	330	329	326
False +	9	13	19	20	21	26	28	30	31	34
<b>Set 6</b>										
True +	187	267	308	335	363	388	405	415	431	444
False -	1004	924	883	856	828	803	786	776	760	747
True -	352	344	339	336	335	334	333	330	329	329
False +	8	16	21	24	25	26	27	30	31	31

**Table 48:** Hit rates for comparison of automatic game to Wizard

	N=1	N=2	N=3	N=4	N=5	N=6	N=7	N=8	N=9	N=10
<b>Set #0</b>										
True +	25.19	33.25	36.94	39.88	42.65	44.16	45.59	47.02	48.03	49.03
False -	74.81	66.75	63.06	60.12	57.35	55.84	54.41	52.98	51.97	50.97
True -	95	94.44	93.33	91.94	91.39	91.11	90.56	90.28	89.72	89.72
False +	5	5.56	6.67	8.06	8.61	8.89	9.44	9.72	10.28	10.28
<b>Set #1</b>										
True +	40.55	49.2	53.99	56.84	59.28	60.96	62.47	63.81	64.57	65.99
False -	59.45	50.8	46.01	43.16	40.72	39.04	37.53	36.19	35.43	34.01
True -	95.28	93.61	92.5	91.11	90.28	88.89	87.5	87.22	86.67	86.11
False +	4.72	6.39	7.5	8.89	9.72	11.11	12.5	12.78	13.33	13.89
<b>Set #2</b>										
True +	27.96	34.01	38.37	41.81	44.58	45.76	47.44	48.53	49.62	50.29
False -	72.04	65.99	61.63	58.19	55.42	54.24	52.56	51.47	50.38	49.71
True -	96.39	94.44	93.06	92.22	91.67	90.83	90.28	90.28	90	89.72
False +	3.61	5.56	6.94	7.78	8.33	9.17	9.72	9.72	10	10.28
<b>Set #3</b>										
True +	42.74	51.89	57.51	60.29	62.55	64.32	65.83	67.51	68.51	69.52
False -	57.26	48.11	42.49	39.71	37.45	35.68	34.17	32.49	31.49	30.48
True -	94.72	93.89	92.22	91.11	89.72	89.17	88.61	87.5	86.67	86.39
False +	5.28	6.11	7.78	8.89	10.28	10.83	11.39	12.5	13.33	13.61
<b>Set #5</b>										
True +	20.15	27.12	32.07	34.17	36.44	38.12	39.63	40.81	41.81	42.91
False -	79.85	72.88	67.93	65.83	63.56	61.88	60.37	59.19	58.19	57.09
True -	97.5	96.39	94.72	94.44	94.17	92.78	92.22	91.67	91.39	90.56
False +	2.5	3.61	5.28	5.56	5.83	7.22	7.78	8.33	8.61	9.44
<b>Set #6</b>										
True +	15.7	22.42	25.86	28.13	30.48	32.58	34.01	34.84	36.19	37.28
False -	84.3	77.58	74.14	71.87	69.52	67.42	65.99	65.16	63.81	62.72
True -	97.78	95.56	94.17	93.33	93.06	92.78	92.5	91.67	91.39	91.39
False +	2.22	4.44	5.83	6.67	6.94	7.22	7.5	8.33	8.61	8.61

## 7.6 Analysis

### 7.6.1 Automatic Game Results

Overall, the best performing configuration was with  $N=10$  for Set#3. Table 49 shows a comparison of the results for Set #3 with N-best  $N = 10$  for three different configurations: comparison of the Wizard’s classification against the hand-labeled ground truth (Wizard/Language), comparison of the Automatic Game classification against the Wizard’s classification (Automatic/Wizard), and comparison between the Automatic Game against the hand-labeled ground truth (Automatic/Language). The Automatic Game achieved a 75.63% accuracy when compared to ground truth (Automatic/Language) and a 73.44% accuracy when compared to the live Wizard’s answers (Automatic/Wizard). Of particular interest is the kind of errors the automatic game made.

The Wizard/Language results show true positive / false negative weighting of 95.45%/4.55% and a true negative / false positive weighting of 84.34%/15.66%. This distribution is interesting because the Wizard gave very few false negatives, but significantly more false positives. The wizard was three times more likely to err in favor of the child than to falsely penalize them. The results are even more telling since there were three times more positive examples than negative. Even with such a relative large number of positives, the false negative rate is quite low.

The Automatic/Language results show a true positive / false negative weighting of

**Table 49:** Summative comparison of results of Set #3 with N-best  $N = 10$ . The table shows a comparison between the Wizard’s live performance and the ground truth language evaluation, comparison between the automatic game and the Wizard’s live performance, and comparison between the automatic game and the ground truth language evaluation.

	<b>Between Wizard &amp; Language</b>	<b>Between Automatic &amp; Wizard</b>	<b>Between Automatic &amp; Language</b>
<b>Metric</b>	<b>Language</b>	<b>Wizard</b>	<b>Language</b>
Accuracy	92.85%	73.44%	75.63%
True +	95.45%	69.52%	71.02%
False -	4.55%	30.48%	28.98%
True -	84.34%	86.39%	90.66%
False +	15.66%	13.61%	9.34%

71.02%/28.98% and a true negative / false positive weighting of 90.66%/9.34. These distributions are the reverse of the Wizard's. The system was about three times more likely to generate a false negative than a false positive. This effect also disproportionately lowered the system accuracy due to the larger number of positive examples.

### 7.6.2 Expanding N

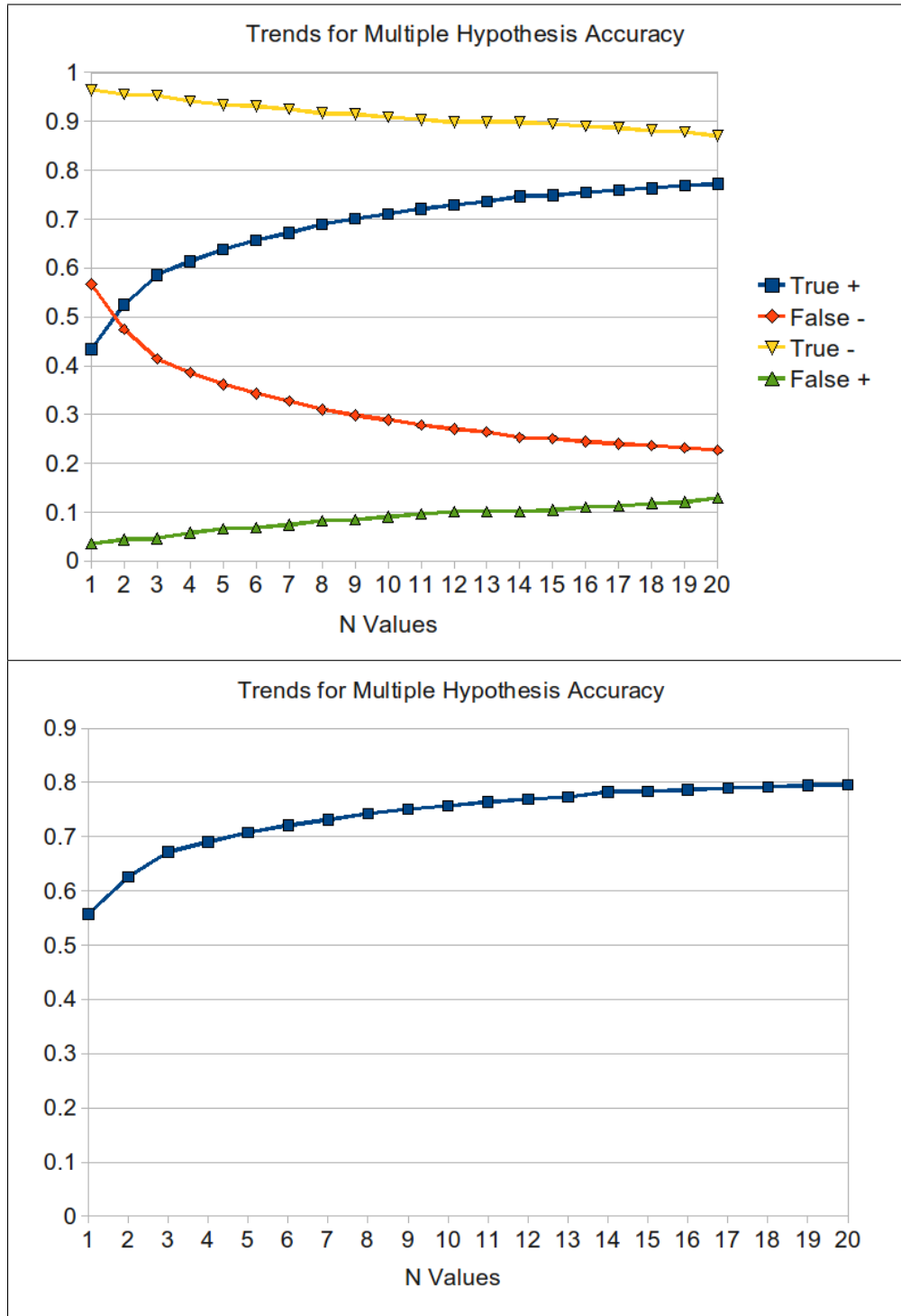
It is worth observing that increasing N has improved the automatic game performance. What are the continued gains in performance for larger N? Figure 45 shows the trend lines for calculating out the automatic game statistics for  $N = 20$  on ground truth. The accuracy trend line in this graph appears to be asymptotic to around 80%. This limitation around 80% is a product of the current limitations of the recognition algorithm.

If our goal is to approximate the Wizard's distribution with a true positive / false negative weighting of 95.45%/4.55% and a true negative / false positive weighting of 84.34%/15.66%, then the trend lines for  $N=20$  show that the limits of the recognition will probably prevent a good match since the true positive / false negative rates approach around 75/25 at  $N = 20$  and appear to be barely increasing. Though the values for  $N = 20$  are closer to our Wizard's profile, they still have room for improvement.

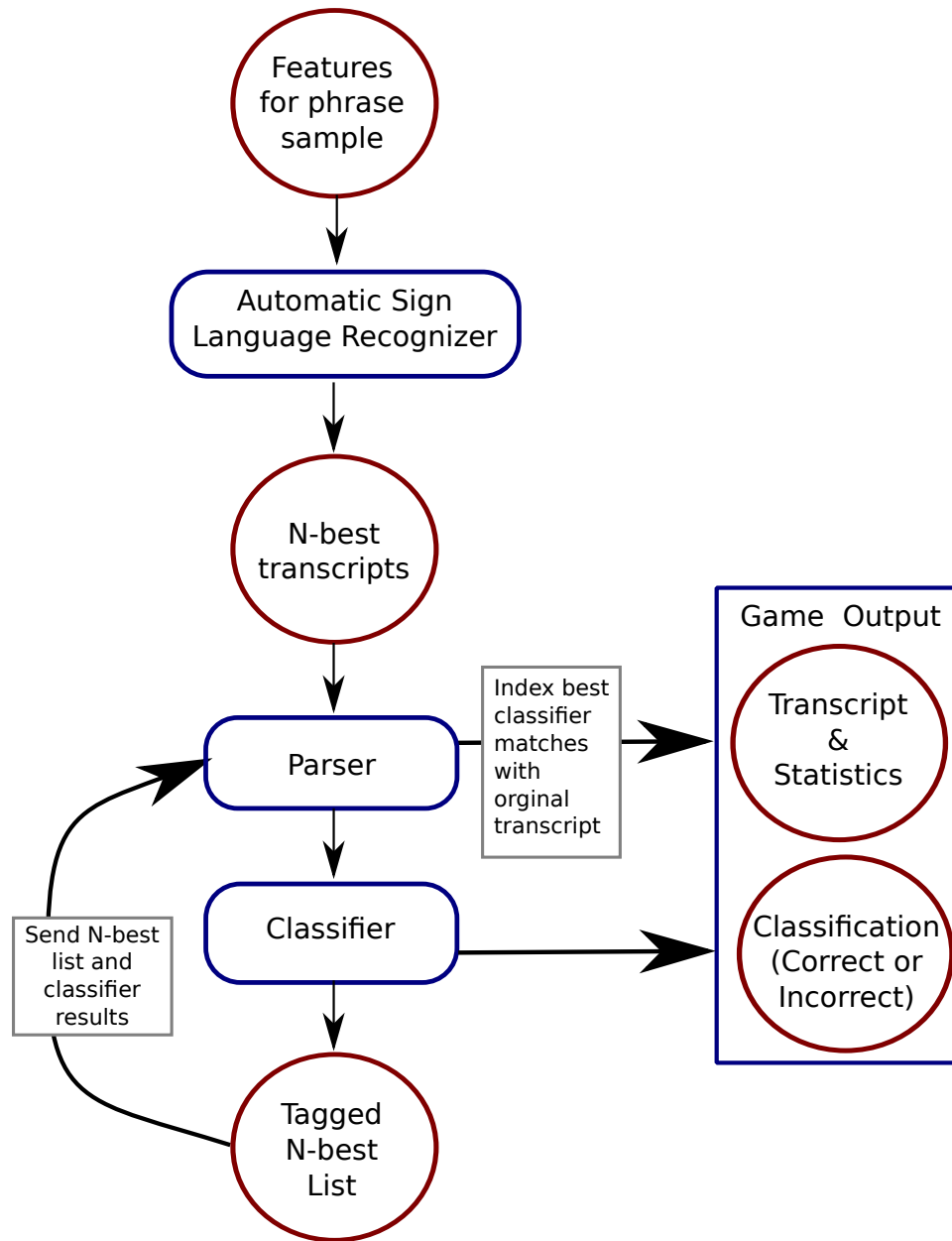
### 7.6.3 Extending Game Response

The application of the Set #3 data labeling scheme provides information on handedness and quality in the phrase transcriptions. N-best lists are frequently used by either applying natural language processing as a filter or as supplementary information for re-scoring and re-ranking the list [112]. Figure 46 shows how the automatic game could be extended to provide feedback to the game about handedness and quality of sign. After the Classifier generates an answer, the N-best lists and Classifier output can be fed back into the Parser.

The Parser would then select the best match transcription based on the Classifier output and the N-best list. In the case that the phrase was selected as correct, the Parser would index the first transcript on the N-best list that has a positive match from the Classifier. This transcript would be the most likely transcript from the N-best list that was classified as correct. In the case that the phrase was classified as incorrect, the top (most likely)



**Figure 45:** Trends for  $N = 20$  experiments: Top graph shows trends for true positive, false negative, true negative, and false positive. Bottom graph shows trends for accuracy.



**Figure 46:** Extending the game responses by integrating the details on handedness and quality of signs (extension of the automatic game diagram in Figure 41)

transcript would be selected from the list.

The selected transcript can then be reported to the game, where it could be used for teacher reports or direct user feedback. The quality can be reported on a sign-by-sign basis, or average quality could be reported. This kind of information could be used to provide direct feedback to the child in the form of customized tutoring for errors or low-quality signs or customized responses to varying quality sign.

### ***7.7 Summary***

I have presented the design and performance of the final components of the automatic game: the Parser and the Classifier. These two components performed excellently on tests using hand-labeled sign transcriptions, with an accuracy of 96.98%. These two new components were combined with the automatic recognition to provide a complete system. The resulting accuracies of the automatic game appear to be limited to approximately 80% accuracy due largely to the limitations of the accuracy with the automatic sign language recognition. In comparison, the Wizard's accuracy was 92.85% when compared to a ground truth language evaluation. The automatic game can use the additional information provided by the labeling scheme Set #3 to further the educational goals of the project. In addition Set #3 improved automatic sign language recognition results and increase the detail of the resulting transcripts.

## CHAPTER VIII

### DISCUSSION AND FUTURE WORK

#### *8.1 Discussion*

Chapter 1 provides an introduction to the material, thesis and research activities and contributions. Chapter 2 explores research related to various aspects of the dissertation including sign language linguistics, automatic speech recognition, and automatic sign language recognition.

Chapter 3 provides a detailed introduction to the CopyCat project and its related corpus. I present a description and time line of the CopyCat project. The current CopyCat system uses a phrase verification system, which is a binary classifier of correct/ incorrect. The verification system compares the signed phrase against the expected phrase for game play. If the probability of a match is high enough the system will consider the answer correct. If the probability is too low the system will classify the answer as incorrect. This system performs best on well-structured signing. Signed phrases with variations in language usage and structure can create problems for the verifier.

##### **8.1.1 Labeling Ontology**

I present the development of an ontology based on three criteria: sign label, handedness, and quality. During the development of the ontology I identified several out-of-vocabulary signs, including signs in Signing Exact English (SEE). Several classes of disfluencies were defined for the data set as well. Handedness was described using information on dominant hand, symmetry, and the number of active hands (one or two). Quality was defined by the labels of GOOD, OK, and BAD.

One of the larger research questions of the CopyCat project has been how to define the content of our Corpus. I used the developed ontology to characterize the data set. These qualitative descriptions provided important information on the distribution of the sign labels, dominant hand usage, and quality. There were 1191 phrases labeled, which

resulted in 9055 segments. Each segment represents a distinct sign, disfluency, or non-sign gesture.

Dominant hand usage was found to vary by child, with right hand dominance being more well defined than left hand dominance. The overall quality of the signing was rated at 94% GOOD, but 4% of the signs were rated as OK and 2% as BAD. In the student-by-student analysis two students performed 99% of their signs at GOOD quality, but 22% of the students had less than 90% GOOD signs.

Additionally, disfluencies were found to have a major impact on the data set. Of the 9055 labeled sign segments, 10.30% of the segments were out-of-vocabulary signs or disfluencies. Of the 1191 labeled phrases, 46.52% of the phrases contained a disfluency and 3.36% of the phrases contained at least one out-of-vocabulary sign.

Finally, an analysis of the language structure used by the children in the data set showed evidence that a traditional structured grammar would be insufficient for our recognition engine. Only 43.07% of the samples conformed to a structured grammar for the correct game phrase. Thus, a structured grammar was not used in the recognition system.

### **8.1.2 Recognition Experiments**

Chapter 5 covers the infrastructure and techniques used for modeling signs and gestures. In Chapter 6 I discuss the application of the labeling ontology to the automatic sign language recognition system. This discussion covers the progression of recognition experiments that results in the final automatic sign language recognition engine used in the game.

I present the automatic sign language recognition research and results. The labeling ontology was used to generate six labeled sets for the recognition experiments. Sets #0, #1, #2, and #3 were used to evaluate the impact of different labeling permutations. Sets #5 and #6 were used as a baseline comparison of previous techniques.

My experimental design used hidden Markov models for recognition and no structured grammar. The features used for modeling were a combination of information from head tracking, hand tracking, and the accelerometers. Within each labeled set I used a single model class per label instance. These models were trained and then used for automatic

recognition using the hand-labeled transcripts as ground truth. The results from each of these recognition tests was used to compare performance.

The consistent best performer was labeled Set #3, which included information on sign label, handedness, and quality. The baseline recognition results using previous methods showed word accuracies of 60.78% and 61.19%. Set #3 resulted in 70.97% word accuracy. A post-processing technique that ignored non-sign activities showed results of 77.22% word accuracy for Set #3. Exploring multiple hypotheses increased the word accuracy to 87.10% for Set #3 using N=10 for N-best recognition.

### 8.1.3 Automatic Game

In Chapter 7 I discuss the application of the ontology in the design of the classifier and parser components for the automatic game. These components are combined with the automatic sign language recognition engine and the automatic game system is evaluated.

I present work on profiling the Wizard and data set, as well as the creation of the automatic game and the resulting game performance. When compared to the hand-labeled transcripts, the Wizard performed at 92.85% accuracy. The Wizard's true positive/ false negative rate was 95.45% / 4.55%. The Wizard's true negative/ false positive rate was 84.34% / 15.66%. Understanding the Wizard's performance gives us a baseline for comparison since the deployments with the Wizard resulted in a positive learning effect. Additionally, a profile of the data set showed a positive bias of 76.53%.

I evaluated the parser performance by using the hand-labeled transcripts as input. The parser performed at 96.98% accuracy. The true positive/ false negative rate was 98.15% / 1.85%. The true negative/ false positive rate was 74.14% / 25.86%. These results showed that the impact of the human judgment-based decisions was minimal and accounted for a loss of 3.02% in accuracy overall.

I evaluated the entire automatic game by using the multiple hypothesis transcripts as input to the parser. Using N=10 for N-best recognition, the results showed that Set #3 (handedness and quality) still performed the best at a 75.63% phrase accuracy for N=10. The previous methods perform at 35.65% and 38.88% phrase accuracy at N=1. At N=10

the previous methods barely cross chance (the 50% line).

I evaluated the convergence trends for the system by testing for N=1 to N=20 for N-best recognition input. The phrase accuracy for Set #3 appears to asymptotic to around 80%. The true negative/ false positive rates approach the Wizard's performance, but the true positive/ false negative results do not appear to be converging to the Wizard's rates. In general, the Wizard is more likely to err in favor of the student (false positive) and the automatic game is more likely to err in against the student (false negative).

The new game architecture integrates the recognition system and parser:

- Automatic sign language recognition generates N-best transcripts
- Parser and classifier pick best match from recognition N-best lists and either accept or reject the phrase
- The best match transcription is passed back to the game and can be used for other tasks
- Information about signing, dominant hand usage, and sign quality can be used for feedback to students and educators

The new automatic game now uses N-best transcripts as input to a parser which outputs a classification of correct or incorrect. This approach uses the infrastructure of a small scale dialogue system to add more flexibility to language usage in the game. The resulting system achieves a performance limit of around 80%, which is an improvement from 35.65% with previous techniques.

Additionally, the new architecture provides a transcript of the children's signing which includes information on signs, disfluencies, non-sign gestures, sign quality, and sign handedness. This information can be used to provide detailed feedback to the children during game play and summative feedback to educators about children's performance. Future incarnations of CopyCat games could use this information for game advances such as variable difficulty and helping children work on trouble signs.

## ***8.2 CopyCat: Integrating a Quality Assessment***

Figure 46 in Chapter 7 shows a diagram of how handedness and quality information from the Set #3 labeling scheme could be integrated into the current game architecture. Section 7.6.3 contains a brief discussion of how the additional information from the Set #3 labeling scheme could be easily integrated into the game architecture to provide more detailed transcripts of the children’s game performance.

Sign quality information could also be used to provide more targeted feedback to children. The system could provide additional feedback and tutoring to students based on quality information from the sign transcripts generated by the classifier. The tutor videos could be extended to have a special review session for signs that the children were consistently performing poorly. Practice sessions within the game could have simple repetition of trouble signs.

Quality information could also provide customization options for teachers. Novice settings might accept signs of lower quality than an expert setting. Video clips could be tagged as good or bad examples for teacher review. One could imagine a class “TV show” that had the good examples for the week, much like posting A+ papers on the bulletin board.

## ***8.3 CopyCat: Improving Feedback for Teachers***

CopyCat is currently deployed in schools by the CopyCat research team. We typically visit schools for 2-4 weeks for language evaluations and game play. Our long term goal is to permanently deploy CopyCat for classroom use. One useful expansion towards this goal is the development of teacher reports containing information about the children’s game play on an individual level and an aggregate level for the class. The addition of quality and handedness information to these reports could provide useful information about children’s progress in the game and language fluency.

## ***8.4 CopyCat: Expanding Game Functionality***

The copycat game currently uses a 19 sign vocabulary for three, four, and five sign phrases. In the future, game expansions could include an expansion of game scenarios that use new

vocabulary. Additionally, language structures other than the *Subject-Preposition-Object* construction could be included. Easier levels could be added to include sessions with color or object identification. Levels of higher difficulty could be added that require more difficult vocabulary and language structures.

The work of Zafrulla et. al. [163, 164] has focused on the automatic sign language verification task using the CopyCat corpus. This technique is currently being used for the CopyCat automatic game. The work in this dissertation provides a dialogue system approach to the CopyCat game. In the future, we can divide the CopyCat game tasks into memory games which use the verification system and tasks that are more descriptive or narrative using the dialogue system.

Until the accuracy of the dialogue system increases, one could imagine an interim system that uses both the verification technique and some of the approaches developed in this dissertation. The verification system could be used to accept or reject a sample, but the dialogue system could provide a second pass for tagging the segmentations provided by the verification system with handedness and quality information.

### ***8.5 CopyCat: Improving the Automatic Game***

The expanded N-best experiments in Section 7.6.2 graph the trends for N-best performance with larger values of N. The trends show that the accuracy limit of the current automatic game is around 80%. The baseline comparison in Section 7.4.5 showed that 3.02% of the samples were decided by a human judgment call that superseded the game transcript. This low error rate means the component in need of improvement is the recognition engine.

The larger set of classes modeled with the Set #3 labeling scheme results in fewer samples per model. Additional signing samples could help us improve our models by providing more training examples. We are continuing test deployments of CopyCat and will be labeling this data for training.

### ***8.6 Discriminative Power of Models***

As discussed in Section 6.5.1 there are several sign confusions, such as BLUE and GREEN, that create a disproportionate amount of errors for the recognition engine. There are several

good techniques for reducing the incidence of the problems. One approach would be to add more discriminative power to our recognition engine by using a technique such as discriminative HMMs [161, 43] or segmental discriminative analysis [160]. Another would be to further refine our features to help provide more information that are relative to these signs. In this case, BLUE and GREEN are minimal pairs that are differentiated by hand shape (see Section 2.1.7 for a linguistics discussion of minimal pairs). Perhaps more detail could be added for features pertaining to hand shape.

### ***8.7 Segmentally Boosted HMMs***

The work of Yin [160] describes a technique for using segmentally boosted HMMs for ASLR. This technique could potentially provide more discriminative power to the models, as well as provide a mechanism for dealing with the problem of insufficient data for some classes. State tying has been used in speech to improve performance on smaller data sets and increase the scalability of recognition systems [43, 161]. Yin’s work combined state tying and selected boosting for ASLR. The modified state typing algorithms developed by Yin may provide some of these benefits to the CopyCat ASLR system.

### ***8.8 Refinement of Ontology***

The non-sign classes (START\_SENTENCE, END\_SENTENCE, SILENCE, GARBAGE, FIDGET, and WAVE) are still fairly general classes. It is possible that a more detailed review of examples of these classes will provide further insight into their structure and usage. Breaking these classes into sub-classes could further improve our recognition rates. The Set #3 ontology provides a basic infrastructure to label signs for the CopyCat game.

Further research questions include

- How does that ontology generalize to other sign-based dialogue systems?
- How would it compare to adult’s signing in an interactive game like CopyCat?
- How does it compare to conversational signing or signing in television shows?

These research questions provide many rich areas for exploration.

## ***8.9 Modeling Signed Languages***

Linguists have shown that signed languages have many common linguistic structures [121]. Several research groups have worked on data sets collected from different signed languages: British Sign Language and ASL [22], Dutch Sign Language (NSL) and ASL [136], and German Sign Language (DGS) and ASL [13, 14]. In the United States both ASL and SEE are used. In the future it is possible that the CopyCat game could provide an interactive sign language game in multiple languages.

Though my ontology is based on a sign language dialogue system, it may provide broader implications for gesture based systems. Many gesture-based systems use an arbitrary set of gestures designed by researchers. Signed languages provide a well-developed natural language of gestures that could be used as a gesture input language for computing systems in the future.

## ***8.10 Final Remarks***

In much of the research community there is the strong desire to create a meaningful artifact that changes the world. Working on the CopyCat project has certainly been a unique opportunity and the experience to have been there from the beginning is nothing short of magical. With much of applied research, there is a constant struggle between answering meaningful research questions and engineering a functional system. The diversity of research stake holders in the CopyCat project has created a unique, truly interdisciplinary research experience that I believe is a rare opportunity.

I feel that Aist and Mostow summarized many of more lively CopyCat project meeting discussions in an excerpt from “Designing Spoken Tutorial Dialogue with Children to Elicit Predictable but Educationally Valuable Responses”

A key question in speech-enabled intelligent tutoring systems is how to design technically feasible, educationally effective spoken tutorial dialogue. This challenge is especially acute with children, for whom speech recognition is particularly difficult. Recognizing speech is hard for computers, children’s speech even harder (even for humans); the feasibility of spoken dialogue with children

depends to a great extent on the predictability of speech.

These twin challenges, predictability on the one hand, and educational effectiveness on the other, are often at odds. Naturalistic dialogue, based on human tutor behavior, draws on the effectiveness of human tutoring behaviors yet automatic speech recognition lacks the accuracy of human speech recognition, especially for children. [4]

It is my sincere hope that this dissertation will help bring us one large step closer to creating this balance by improving our ability to predict responses in a system we have designed to be educational. Maybe our system will help improve the communication skills and perhaps the lives of a some children out there. As a eccentric, but wise, man once said “You have 50 years. How do you want to change the world?” Of course, slightly less wise graduate students have also replied with answers like “square root of green?”, “the delicious, delicious cheese theorem?”, and “huh?” - but they probably needed some sleep.

## CHAPTER IX

### CONCLUSION

The CopyCat project is an interdisciplinary effort to create a set of computer-aided language learning tools for deaf children. CopyCat uses automatic sign language recognition in order to allow children to sign to characters within game play. The CopyCat data set is unique corpora in the field of automatic sign language. The Phase Two deployment subset of the CopyCat data has signing examples of continuous signing from 19 children, signing 60 different phrases, using a game vocabulary of 19 ASL signs. During game play the children used 23 different signs, two of which are from SEE. CopyCat is unusual as a automatic sign language recognition (ASLR) project since it uses sign data from children. Recent survey papers show only two ASLR projects using more than 5 signers [59, 74, 105, 137].

In this dissertation I have presented an analysis of the data of children signing to video game characters collected during the CopyCat Phase 2 deployment. From this data, I have created a labeling ontology based on gesture labels, sign handedness, and quality. I used this ontology to characterize the contents of the data set. This analysis shows a wide variation in the children's dominant hand usage, as well the consistent usage of several out-of-game signs. The children's signing was predominantly ASL, but contained usage of several SEE signs and grammatical structures. The children exhibited several disfluencies, the most common being extended silences and conversational repairs.

This labeling ontology was used to create test sets for recognition experiments. My initial hypothesis was that I could use the quality labels to prune samples for modeling. This technique did not provide a significant gain in recognition rates, but other experiments show that using the quality labels for separate models performed well (Set #1 and Set #3). Including information on handedness (Set #2 and Set #3) resulted in a consistent gain in recognition rates. The test set with the best performance was also the test set with the most information in its labeling scheme (Set #3). The use of N-best recognition improved

the recognition rates with all schemes, but again, the set with the most information was the top performer. Post-processing Set #3 achieved a word accuracy of 87.10% and a sentence correctness of 48.53% for N-best recognition where  $N = 10$ .

Information from the data characterization was combined with the N-best hypothesis recognition engine to design the final classifier for the automatic game. The automatic game combined a multiple hypothesis language parser to classify samples as correct or incorrect. The Wizard was evaluated at 92.85% accuracy when compared to a retrospective ground truth labeling. The classifier component of the automatic game was evaluated at 96.98% when using the hand-labeled transcripts as input. When the N-best recognition engine was used as input, the automatic game achieved a accuracy of 75.63%. A comparison of the Wizard's performance profile and the profile of the automatic game showed the Wizard tended to err toward false positives, while the automatic game tended to err with false negatives. A plot of the results of N-best for  $N = 20$  showed that the limit for performance with the developed game is probably around 80%. In light of this, future work should focus on improvements of the recognition engine.

In this dissertation I demonstrated that modeling the variations and disfluencies that occur in a casual signing context can improve the accuracy of a sign recognition system for an ASL practice tool and the performance of an automatic game. Specifically, I have created an ontology for labeling variations and disfluencies in the CopyCat data set. I have used this ontology to improve our automatic sign language recognition for CopyCat, as well as expanded the language processing for the game. The resulting game provides additional information about sign handedness and quality that can be integrated into the game in order to provide more detailed feedback to users and educators.

## APPENDIX A

### TOOL OVERVIEW

#### A.1 HTK

The Hidden Markov Model Toolkit (HTK) was developed at the Speech, Vision and Robotics group at the Cambridge University Engineering Department as a toolkit for building and using Hidden Markov models for speech recognition research. HTK uses speech data to train models, which can then be used for recognition. HTK consists of a library of tools primarily used for speech recognition, though it has gained popularity for non-speech applications. It is free to download, use and modify HTK, but it is not allowed to be redistributed. Applications and models can be built using HTK, but cannot distribute HTK source with a product. HTK is available on developed for use with Unix, but is also available for Windows and Mac systems. The tools that I have used in training are described in Table 50. The tools that I have used in testing are described in Table 51.

**Table 50:** Reference for HTK training tools

<b>HTK Tool</b>	<b>Input</b>	<b>Result</b>
MakeProtoHMMSet	<i>dictionary, HMM topology</i> definitions	Generates a HMM with a specified <i>topology</i> for each token in the <i>dictionary</i>
HInit	<i>training samples, HMM</i>	Initializes the HMM by taking the global averages of the <i>training samples</i>
HRest	<i>training samples, HMM</i>	Trains the HMM in isolation using <i>training samples</i>
HERest	<i>training samples, HMM</i>	Trains the HMM in context using <i>training samples</i>

**Table 51:** Reference for HTK testing tools

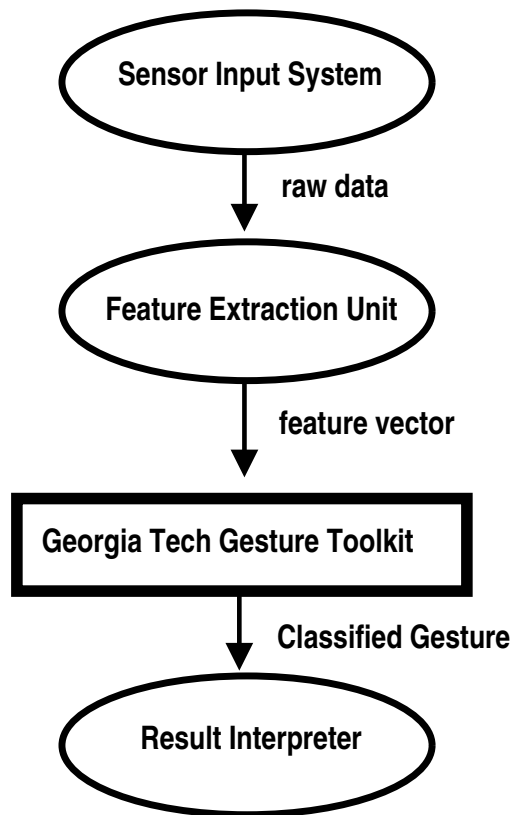
<b>HTK Tool</b>	<b>Input</b>	<b>Result</b>
HVite	<i>HMM, dictionary, grammar, testing samples</i>	Uses HMMs and grammar to generate a <i>transcript</i> of the recognized <i>testing samples</i>
HResults	Recognition <i>transcription</i> , Ground truth <i>transcription</i>	Compares the recognition results with labeled ground truth.

## A.2 *GT<sup>2</sup>k*

The Georgia Tech Gesture Toolkit (*GT<sup>2</sup>k*) provides a publicly available toolkit for developing gesture-based recognition systems [151]. The toolkit allows easy development of the gesture recognition component of larger systems. Figure 47 shows the integration of *GT<sup>2</sup>k* into such a system. First, sensors such as video cameras or accelerometers gather data about the gesture being performed. This sensor data can be processed to ascertain the salient characteristics, known as features. The *Data Generator* collects this data and provides the features that are used by *GT<sup>2</sup>k* components to perform training and recognition. The results returned by *GT<sup>2</sup>k* are considered by the *Results Interpreter* and acted upon based on the needs of the application.

The speech-recognition community has invested significant resources into development of recognition technology. The Hidden Markov Model Toolkit (HTK) [161], an open source HMM toolkit, was developed for speech recognition applications. *GT<sup>2</sup>k* serves as a bridge between the user and HTK services by abstracting away the lower level speech-specific functionality and allowing the user to leverage the full power of HTK's HMM manipulation tools. *GT<sup>2</sup>k* allows the gesture-recognition community to benefit from the speech-recognition community's research by providing a tool powerful enough to satisfy the needs of people versed in HMM literature but simple enough to be used by novices with little or no experience with HMM techniques.

*GT<sup>2</sup>k* allows researchers to focus on developing systems that use gesture recognition and the research surrounding those projects, instead of devoting time to recreate existing gesture recognition technology. *GT<sup>2</sup>k* abstracts the lower level details of the pattern recognition



**Figure 47:** GT<sup>2</sup>k interaction with application components (Image used from Westeyn, Brashear, Atrash, and Starner “Georgia Tech Gesture Toolkit: Supporting Experiments in Gesture Recognition” [151])

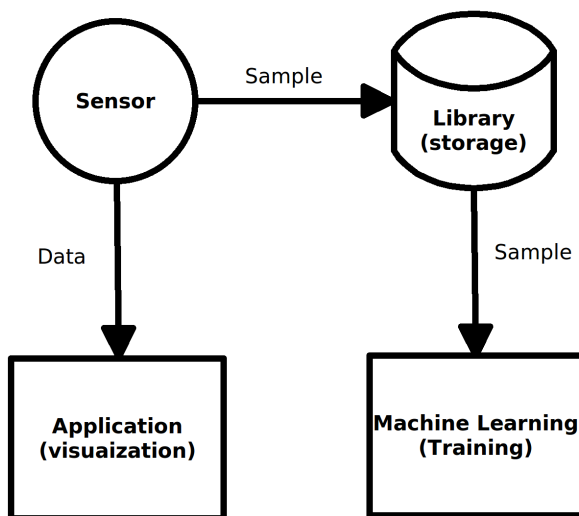
process and allows users to focus instead on high level gesture recognition concepts by providing a suite of configurable tools. Appropriate applications for GT<sup>2</sup>k are systems which utilize discrete gestures, such as sign language, handwriting, facial gestures, full body activities, and issuing robot commands. GT<sup>2</sup>k is not designed for the creation of tracking devices such as those that might be used for controlling a mouse [25]. This toolkit may be of interest to researchers in the areas of human–computer interaction, assistive technologies, robotics, and other fields involving gesture recognition.

GT<sup>2</sup>k provides a user with tools for preparation, training, validation, and recognition using HMMs for gesture–based applications. Preparation requires that the user design gesture models, determine an appropriate grammar, and provide labeled examples of the gestures to be trained. Training uses information from the preparation phase to train models of each gesture. Validation evaluates the potential performance of the overall system. Recognition uses the trained models to classify new data. At this point, GT<sup>2</sup>k assumes data is being provided by a *Data Generator*, such as a camera, microphone, or accelerometer, in the form of a feature vector. The resulting GT<sup>2</sup>k classification is then handled by a *Results Interpreter* as appropriate for the application.

### ***A.3 GART Overview***

The Gesture and Activity Recognition Toolkit (GART) is a user interface toolkit. It is designed to provide a high level interface to the machine learning process facilitating the building of gesture recognition applications [89]. The toolkit consists of an abstract interface to the machine learning algorithms (training and recognition), several example sensors and a library for samples.

To build a gesture based application using GART, the programmer first selects the sensor she will use to capture information about the gesture. We currently support three basic sensors in our toolkit: a mouse (or pointing device), a set of Bluetooth accelerometers, and a camera sensor. Once a sensor is selected, the programmer builds an application that can be used to collect training data. This program can be either a special mode in the final application being built, or an application tailored just for data collection. Finally,

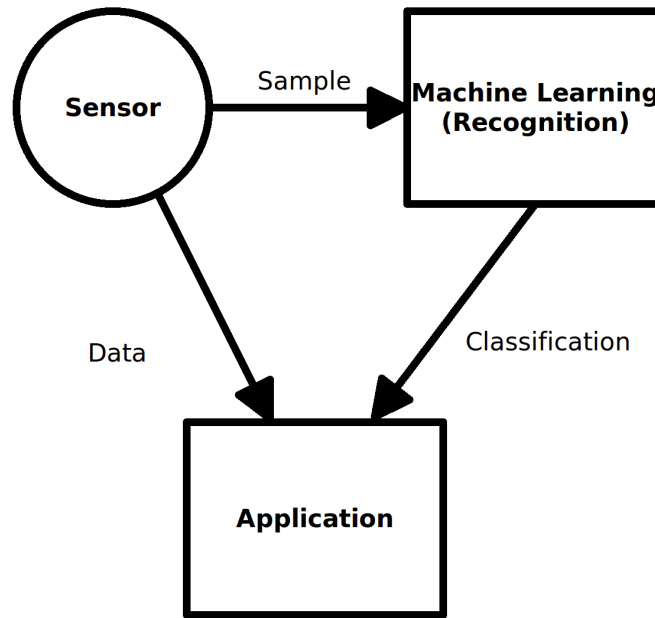


**Figure 48:** Data collection using GART (Image used from Lyons, Brashear, Westeyn, Kim and Starner “GART: The Gesture and Activity Recognition Toolkit” [89])

the programmer instantiates the base classes from the toolkit (encapsulating the machine learning algorithms, and library) and sets up the call-backs between them for data collection or recognition. The remainder of the programmer’s coding effort can then be devoted to building the actual application of interest and using the gesture recognition results as desired.

The toolkit is composed of three main components: *Sensors*, *Library*, and *Machine Learning*. *Sensors* collect data from hardware and may provide post-processing. The *Library* stores the data and provides a portable format for sharing data sets. The *Machine Learning* component encapsulates the training and recognition algorithms. Data is passed from the sensor and machine learning components to other objects through call-backs.

The flow of data through the system for data collection involves the above three toolkit components and the application (Figure 48). A sensor object collects data from the physical sensors and distributes it. The sensor will likely send raw data to the application for visualization as streaming video, graphs, or for other displays. The sensor also bundles a set of data with its labeling information into a sample. The sample is sent to the library where it stored for later use. Finally, the machine learning component can pull data from the library and use it to train the models for recognition.



**Figure 49:** Recognition using GART (Image used from Lyons, Brashear, Westeyn, Kim and Starner “GART: The Gesture and Activity Recognition Toolkit” [89])

Figure 49 shows the data flow for a recognition application. As before, the sensor can send raw data to the application for visualization or user feedback. The sensor also sends samples to the machine learning component for recognition, and recognition results are sent to the application.

## APPENDIX B

### LABEL FREQUENCIES IN DATA

Tables show the full listing of the frequency distributions of labels, per scheme, across the data set.

Table 52: Tallies of the classes for Set #0. Shows the number of examples for each label, as well as the number of unique signers for that sign

Label	Fall		Spring I		Spring II		All	
	Examples	Signers	Examples	Signers	Examples	Signers	Examples	Signers
alligator	109	5	63	5	166	13	338	18
bed	59	5	25	5	103	13	187	18
behind	47	5	20	5	95	13	162	18
black	88	5	67	5	111	12	266	17
blue	190	5	123	5	222	14	535	18
box	58	5	34	5	86	13	178	18
cat	116	5	85	5	160	13	361	18
chair	62	4	34	4	111	12	207	16
chair-Chand	21	2	6	1	10	3	37	5
end-sentence	344	5	221	5	574	14	1139	18
erase-wave	29	4	8	4	38	11	75	14
fidget	8	3	8	3	34	6	50	9

Table continued on next page

Table 52 – continued from previous page

Label	Fall		Spring I		Spring II		All	
	Examples	Signers	Examples	Signers	Examples	Signers	Examples	Signers
flowers	101	5	67	5	131	14	299	18
garbage	53	4	31	5	64	12	148	17
green	168	5	153	5	224	13	545	18
in	121	5	71	5	156	14	348	18
is	0	0	0	0	7	1	7	1
on	114	5	75	5	166	13	355	18
orange	123	5	75	5	155	12	353	17
silence	295	5	86	5	226	13	607	18
snake	122	5	58	5	146	14	326	18
spider	105	5	57	5	162	13	324	18
start-sentence	365	5	208	5	558	14	1131	18
the	1	1	10	2	32	5	43	7
under	151	5	77	5	206	12	434	17
wagon	36	4	20	5	64	13	120	18
wall	64	5	44	5	96	12	204	17
white	95	5	90	5	88	13	273	17
wrong	1	1	0	0	2	2	3	3

Table 53: Tallies of the classes for Set #1. Shows the number of examples for each label, as well as the number of unique signers for that sign

Label	Fall		Spring I		Spring II		All	
	Examples	Signers	Examples	Signers	Examples	Signers	Examples	Signers
alligator-both-left	17	1	14	1	35	5	66	6
alligator-both-right	92	4	49	5	131	12	272	17
bed-both-left	13	3	11	4	60	10	84	14
bed-both-right	41	4	10	2	22	5	73	9
bed-left	5	1	4	2	1	1	10	3
bed-right	0	0	0	0	20	6	20	6
behind-both-left	6	1	4	1	28	6	38	7
behind-both-right	41	5	16	4	67	10	124	15
black-left	14	2	8	1	34	6	56	8
black-right	74	4	59	5	77	10	210	15
blue-left	41	3	24	3	71	8	136	11
blue-right	149	4	99	5	151	12	399	17
box-both-left	8	1	4	1	38	10	50	11
box-both-right	50	5	30	5	48	11	128	16
cat-both	70	4	62	5	44	6	176	11
cat-left	12	2	6	1	50	7	68	9
cat-right	34	3	17	4	66	10	117	14
chair-both-left	1	1	0	0	43	6	44	7
chair-both-right	61	4	34	4	68	9	163	13

Table continued on next page

Table 53 – continued from previous page

Label	Fall		Spring I		Spring II		All	
	Examples	Signers	Examples	Signers	Examples	Signers	Examples	Signers
chair-Chand-both-left	20	1	6	1	10	3	36	4
chair-Chand-both-right	1	1	0	0	0	0	1	1
end-sentence-right	344	5	221	5	574	14	1139	18
erase-wave-both	7	2	2	1	7	6	16	9
erase-wave-left	8	1	0	0	11	5	19	5
erase-wave-right	14	3	6	3	20	6	40	9
fidget-right	8	3	8	3	34	6	50	9
flowers-left	26	3	12	1	44	9	82	11
flowers-right	75	4	55	5	87	11	217	16
garbage-right	53	4	31	5	64	12	148	17
green-left	31	2	27	1	79	7	137	9
green-right	137	4	126	5	145	12	408	17
in-both-left	37	1	11	1	49	7	97	7
in-both-right	84	4	60	5	107	12	251	17
is-right	0	0	0	0	7	1	7	1
on-both-left	24	1	14	1	48	7	86	8
on-both-right	90	5	61	5	118	12	269	17
orange-left	38	3	15	2	44	7	97	10
orange-right	85	4	60	4	111	11	256	15
silence-right	295	5	86	5	226	13	607	18
snake-left	39	2	8	1	46	8	93	9
snake-right	83	4	50	5	100	12	233	17

Table continued on next page

Table 53 – continued from previous page

Label	Fall		Spring I		Spring II		All	
	Examples	Signers	Examples	Signers	Examples	Signers	Examples	Signers
spider-both-left	19	1	11	2	66	6	96	8
spider-both-right	86	4	46	4	96	12	228	16
start-sentence-right	365	5	208	5	558	14	1131	18
the-left	0	0	0	0	1	1	1	1
the-right	1	1	10	2	31	5	42	7
under-both-left	35	1	14	1	69	7	118	8
under-both-right	116	5	63	4	137	9	316	14
wagon-left	19	2	5	2	12	6	36	8
wagon-right	17	4	15	4	52	10	84	15
wall-both	64	5	44	5	96	12	204	17
white-left	24	2	11	2	30	7	65	9
white-right	71	5	79	5	58	11	208	16
wrong-left	0	0	0	0	1	1	1	1
wrong-right	1	1	0	0	1	1	2	2

Table 54: Tallies of the classes for Set #2. Shows the number of examples for each label, as well as the number of unique signers for that sign

Label	Fall		Spring I		Spring II		All	
	Examples	Signers	Examples	Signers	Examples	Signers	Examples	Signers
alligator-bad	8	3	2	2	10	5	20	8
alligator-good	85	5	60	5	131	12	276	17
alligator-ok	16	4	1	1	25	8	42	12
bed-bad	1	1	0	0	4	3	5	4
bed-good	58	5	25	5	90	13	173	18
bed-ok	0	0	0	0	9	5	9	5
behind-bad	0	0	1	1	4	3	5	4
behind-good	47	5	19	5	80	13	146	18
behind-ok	0	0	0	0	11	5	11	5
black-bad	0	0	3	1	2	2	5	3
black-good	81	5	61	5	102	12	244	17
black-ok	7	4	3	2	7	5	17	9
blue-bad	5	2	0	0	7	4	12	6
blue-good	181	5	113	5	193	13	487	18
blue-ok	4	2	10	5	22	10	36	14
box-bad	2	2	2	1	2	2	6	5
box-good	46	5	30	5	78	13	154	18
box-ok	10	4	2	2	6	5	18	9
cat-bad	7	5	2	2	14	4	23	9

Table continued on next page

Table 54 – continued from previous page

Label	Fall		Spring I		Spring II		All	
	Examples	Signers	Examples	Signers	Examples	Signers	Examples	Signers
cat-good	104	5	75	5	133	13	312	18
cat-ok	5	3	8	3	13	5	26	9
chair-bad	1	1	0	0	3	2	4	3
chair-Chand-good	20	2	6	1	9	2	35	4
chair-Chand-ok	1	1	0	0	1	1	2	2
chair-good	61	4	34	4	93	12	188	16
chair-ok	0	0	0	0	15	7	15	7
end-sentence-bad	0	0	0	0	2	2	2	2
end-sentence-good	344	5	221	5	569	14	1134	18
end-sentence-ok	0	0	0	0	3	3	3	3
erase-wave-good	26	3	8	4	37	11	71	14
erase-wave-ok	3	3	0	0	1	1	4	4
fidget-good	8	3	8	3	34	6	50	9
flowers-bad	1	1	1	1	3	3	5	5
flowers-good	97	5	66	5	122	14	285	18
flowers-ok	3	3	0	0	6	6	9	9
garbage-bad	0	0	0	0	1	1	1	1
garbage-good	53	4	31	5	63	12	147	17
green-bad	4	2	3	2	21	7	28	9
green-good	156	5	145	5	175	13	476	18
green-ok	8	3	5	3	28	10	41	14
in-bad	1	1	0	0	6	5	7	6

Table continued on next page

Table 54 – continued from previous page

Label	Fall		Spring I		Spring II		All	
	Examples	Signers	Examples	Signers	Examples	Signers	Examples	Signers
in-good	118	5	71	5	142	13	331	17
in-ok	2	2	0	0	8	5	10	7
is-good	0	0	0	0	6	1	6	1
is-ok	0	0	0	0	1	1	1	1
on-bad	2	2	1	1	2	1	5	4
on-good	110	5	74	5	157	13	341	18
on-ok	2	2	0	0	7	5	9	7
orange-bad	2	2	0	0	2	2	4	4
orange-good	116	5	72	5	144	12	332	17
orange-ok	5	2	3	2	9	5	17	7
silence-good	295	5	86	5	226	13	607	18
snake-bad	5	3	3	2	2	2	10	5
snake-good	116	5	52	5	137	14	305	18
snake-ok	1	1	3	1	7	6	11	8
spider-bad	1	1	0	0	7	5	8	6
spider-good	100	5	54	5	146	13	300	18
spider-ok	4	2	3	2	9	8	16	11
start-sentence-good	365	5	208	5	553	14	1126	18
start-sentence-ok	0	0	0	0	5	4	5	4
the-bad	0	0	0	0	1	1	1	1
the-good	1	1	10	2	29	4	40	6
the-ok	0	0	0	0	2	1	2	1

Table continued on next page

Table 54 – continued from previous page

Label	Fall		Spring I		Spring II		All	
	Examples	Signers	Examples	Signers	Examples	Signers	Examples	Signers
under-bad	0	0	0	0	6	3	6	3
under-good	142	5	77	5	182	12	401	17
under-ok	9	4	0	0	18	7	27	11
wagon-bad	2	2	0	0	4	2	6	4
wagon-good	34	4	20	5	53	13	107	18
wagon-ok	0	0	0	0	7	5	7	5
wall-bad	4	1	2	1	1	1	7	2
wall-good	60	5	42	5	94	12	196	17
wall-ok	0	0	0	0	1	1	1	1
white-bad	3	3	1	1	1	1	5	4
white-good	81	5	84	5	85	13	250	17
white-ok	11	3	5	3	2	2	18	6
wrong-good	1	1	0	0	2	2	3	3

Table 55: Tallies of the classes for Set #3. Shows the number of examples for each label, as well as the number of unique signers for that sign

Label	Fall		Spring I		Spring II		All	
	Examples	Signers	Examples	Signers	Examples	Signers	Examples	Signers
alligator-both-left-bad	1	1	1	1	3	2	5	3
alligator-both-left-good	12	1	13	1	31	4	56	5
alligator-both-left-ok	4	1	0	0	1	1	5	2
alligator-both-right-bad	7	2	1	1	7	4	15	6
alligator-both-right-good	73	4	47	5	100	11	220	16
alligator-both-right-ok	12	3	1	1	24	7	37	10
bed-both-left-bad	0	0	0	0	2	2	2	2
bed-both-left-good	13	3	11	4	55	10	79	14
bed-both-left-ok	0	0	0	0	3	2	3	2
bed-both-right-bad	1	1	0	0	2	1	3	2
bed-both-right-good	40	4	10	2	15	4	65	8
bed-both-right-ok	0	0	0	0	5	3	5	3
bed-left-good	5	1	4	2	1	1	10	3
bed-right-good	0	0	0	0	19	6	19	6
bed-right-ok	0	0	0	0	1	1	1	1
behind-both-left-bad	0	0	0	0	3	2	3	2
behind-both-left-good	6	1	4	1	22	6	32	7
behind-both-left-ok	0	0	0	0	3	3	3	3
behind-both-right-bad	0	0	1	1	1	1	2	2

Table continued on next page

Table 55 – continued from previous page

Label	Fall		Spring I		Spring II		All	
	Examples	Signers	Examples	Signers	Examples	Signers	Examples	Signers
behind-both-right-good	41	5	15	4	58	10	114	15
behind-both-right-ok	0	0	0	0	8	2	8	2
black-left-bad	0	0	0	0	1	1	1	1
black-left-good	13	2	8	1	31	6	52	8
black-left-ok	1	1	0	0	2	2	3	3
black-right-bad	0	0	3	1	1	1	4	2
black-right-good	68	4	53	5	71	10	192	15
black-right-ok	6	3	3	2	5	3	14	6
blue-left-bad	3	1	0	0	2	2	5	3
blue-left-good	38	3	21	2	61	6	120	9
blue-left-ok	0	0	3	2	8	5	11	6
blue-right-bad	2	1	0	0	5	3	7	4
blue-right-good	143	4	92	5	132	12	367	17
blue-right-ok	4	2	7	4	14	6	25	10
box-both-left-bad	0	0	1	1	2	2	3	3
box-both-left-good	6	1	2	1	34	10	42	11
box-both-left-ok	2	1	1	1	2	2	5	3
box-both-right-bad	2	2	1	1	0	0	3	3
box-both-right-good	40	5	28	4	44	11	112	16
box-both-right-ok	8	3	1	1	4	3	13	6
cat-both-bad	3	2	2	2	3	1	8	4
cat-both-good	62	4	57	5	41	6	160	11

Table continued on next page

Table 55 – continued from previous page

Label	Fall		Spring I		Spring II		All	
	Examples	Signers	Examples	Signers	Examples	Signers	Examples	Signers
cat-both-ok	5	3	3	2	0	0	8	4
cat-left-bad	2	1	0	0	5	3	7	4
cat-left-good	10	2	3	1	41	7	54	9
cat-left-ok	0	0	3	1	4	2	7	3
cat-right-bad	2	2	0	0	6	2	8	4
cat-right-good	32	2	15	4	51	10	98	14
cat-right-ok	0	0	2	1	9	3	11	4
chair-both-left-bad	0	0	0	0	2	1	2	1
chair-both-left-good	1	1	0	0	38	5	39	6
chair-both-left-ok	0	0	0	0	3	3	3	3
chair-both-right-bad	1	1	0	0	1	1	2	2
chair-both-right-good	60	4	34	4	55	8	149	12
chair-both-right-ok	0	0	0	0	12	5	12	5
chair-Chand-both-left-good	19	1	6	1	9	2	34	3
chair-Chand-both-left-ok	1	1	0	0	1	1	2	2
chair-Chand-both-right-good	1	1	0	0	0	0	1	1
end-sentence-right-bad	0	0	0	0	2	2	2	2
end-sentence-right-good	344	5	221	5	569	14	1134	18
end-sentence-right-ok	0	0	0	0	3	3	3	3
erase-wave-both-good	7	2	2	1	6	5	15	8
erase-wave-both-ok	0	0	0	0	1	1	1	1
erase-wave-left-good	7	1	0	0	11	5	18	5

Table continued on next page

Table 55 – continued from previous page

Label	Fall		Spring I		Spring II		All	
	Examples	Signers	Examples	Signers	Examples	Signers	Examples	Signers
erase-wave-left-ok	1	1	0	0	0	0	1	1
erase-wave-right-good	12	2	6	3	20	6	38	9
erase-wave-right-ok	2	2	0	0	0	0	2	2
fidget-right-good	8	3	8	3	34	6	50	9
flowers-left-bad	1	1	0	0	0	0	1	1
flowers-left-good	24	3	12	1	43	9	79	11
flowers-left-ok	1	1	0	0	1	1	2	2
flowers-right-bad	0	0	1	1	3	3	4	4
flowers-right-good	73	4	54	5	79	10	206	15
flowers-right-ok	2	2	0	0	5	5	7	7
garbage-right-bad	0	0	0	0	1	1	1	1
garbage-right-good	53	4	31	5	63	12	147	17
green-left-bad	2	1	1	1	8	4	11	5
green-left-good	23	2	25	1	63	6	111	8
green-left-ok	6	1	1	1	8	4	15	5
green-right-bad	2	1	2	2	13	5	17	7
green-right-good	133	4	120	5	112	12	365	17
green-right-ok	2	2	4	2	20	8	26	11
in-both-left-bad	1	1	0	0	2	2	3	3
in-both-left-good	35	1	11	1	43	6	89	6
in-both-left-ok	1	1	0	0	4	1	5	2
in-both-right-bad	0	0	0	0	4	3	4	3

Table continued on next page

Table 55 – continued from previous page

Label	Fall		Spring I		Spring II		All	
	Examples	Signers	Examples	Signers	Examples	Signers	Examples	Signers
in-both-right-good	83	4	60	5	99	11	242	16
in-both-right-ok	1	1	0	0	4	4	5	5
is-right-good	0	0	0	0	6	1	6	1
is-right-ok	0	0	0	0	1	1	1	1
on-both-left-bad	0	0	1	1	2	1	3	2
on-both-left-good	23	1	13	1	44	7	80	8
on-both-left-ok	1	1	0	0	2	2	3	3
on-both-right-bad	2	2	0	0	0	0	2	2
on-both-right-good	87	5	61	5	113	12	261	17
on-both-right-ok	1	1	0	0	5	4	6	5
orange-left-bad	1	1	0	0	0	0	1	1
orange-left-good	37	3	14	1	42	6	93	9
orange-left-ok	0	0	1	1	2	2	3	3
orange-right-bad	1	1	0	0	2	2	3	3
orange-right-good	79	4	58	4	102	10	239	14
orange-right-ok	5	2	2	1	7	5	14	7
silence-right-good	295	5	86	5	226	13	607	18
snake-left-bad	2	1	2	1	0	0	4	1
snake-left-good	37	2	5	1	43	8	85	9
snake-left-ok	0	0	1	1	3	3	4	4
snake-right-bad	3	2	1	1	2	2	6	4
snake-right-good	79	4	47	5	94	11	220	16

Table continued on next page

Table 55 – continued from previous page

Label	Fall		Spring I		Spring II		All	
	Examples	Signers	Examples	Signers	Examples	Signers	Examples	Signers
snake-right-ok	1	1	2	1	4	4	7	6
spider-both-left-bad	0	0	0	0	5	3	5	3
spider-both-left-good	17	1	11	2	55	6	83	8
spider-both-left-ok	2	1	0	0	6	5	8	6
spider-both-right-bad	1	1	0	0	2	2	3	3
spider-both-right-good	83	4	43	4	91	12	217	16
spider-both-right-ok	2	1	3	2	3	3	8	5
start-sentence-right-good	365	5	208	5	553	14	1126	18
start-sentence-right-ok	0	0	0	0	5	4	5	4
the-left-good	0	0	0	0	1	1	1	1
the-right-bad	0	0	0	0	1	1	1	1
the-right-good	1	1	10	2	28	4	39	6
the-right-ok	0	0	0	0	2	1	2	1
under-both-left-bad	0	0	0	0	1	1	1	1
under-both-left-good	30	1	14	1	62	6	106	7
under-both-left-ok	5	1	0	0	6	3	11	4
under-both-right-bad	0	0	0	0	5	3	5	3
under-both-right-good	112	5	63	4	120	9	295	14
under-both-right-ok	4	3	0	0	12	4	16	7
wagon-left-bad	2	2	0	0	1	1	3	3
wagon-left-good	17	2	5	2	10	5	32	7
wagon-left-ok	0	0	0	0	1	1	1	1

Table continued on next page

Table 55 – continued from previous page

Label	Fall		Spring I		Spring II		All	
	Examples	Signers	Examples	Signers	Examples	Signers	Examples	Signers
wagon-right-bad	0	0	0	0	3	1	3	1
wagon-right-good	17	4	15	4	43	10	75	15
wagon-right-ok	0	0	0	0	6	4	6	4
wall-both-bad	4	1	2	1	1	1	7	2
wall-both-good	60	5	42	5	94	12	196	17
wall-both-ok	0	0	0	0	1	1	1	1
white-left-bad	1	1	0	0	0	0	1	1
white-left-good	19	2	11	2	30	7	60	9
white-left-ok	4	1	0	0	0	0	4	1
white-right-bad	2	2	1	1	1	1	4	3
white-right-good	62	5	73	5	55	10	190	15
white-right-ok	7	3	5	3	2	2	14	6
wrong-left-good	0	0	0	0	1	1	1	1
wrong-right-good	1	1	0	0	1	1	2	2

Table 56: Tallies of the classes for Set #5. Shows the number of examples for each label, as well as the number of unique signers for that sign

Label	Fall		Spring I		Spring II		All	
	Examples	Signers	Examples	Signers	Examples	Signers	Examples	Signers
alligator	109	5	63	5	166	13	338	18
bed	59	5	25	5	103	13	187	18
behind	47	5	20	5	95	13	162	18
black	88	5	67	5	111	12	266	17
blue	190	5	123	5	222	14	535	18
box	58	5	34	5	86	13	178	18
cat	116	5	85	5	160	13	361	18
chair	83	5	40	5	121	12	244	17
end-sentence	344	5	221	5	574	14	1139	18
erase-wave	29	4	8	4	38	11	75	14
fidget	8	3	8	3	34	6	50	9
flowers	101	5	67	5	131	14	299	18
garbage	53	4	31	5	64	12	148	17
green	168	5	153	5	224	13	545	18
in	121	5	71	5	156	14	348	18
is	0	0	0	0	7	1	7	1
on	114	5	75	5	166	13	355	18
orange	123	5	75	5	155	12	353	17
silence	295	5	86	5	226	13	607	18

Table continued on next page

Table 56 – continued from previous page

Label	Fall		Spring I		Spring II		All	
	Examples	Signers	Examples	Signers	Examples	Signers	Examples	Signers
snake	122	5	58	5	146	14	326	18
spider	105	5	57	5	162	13	324	18
start-sentence	365	5	208	5	558	14	1131	18
the	1	1	10	2	32	5	43	7
under	151	5	77	5	206	12	434	17
wagon	36	4	20	5	64	13	120	18
wall	64	5	44	5	96	12	204	17
white	95	5	90	5	88	13	273	17
wrong	1	1	0	0	2	2	3	3

Table 57: Tallies of the classes for Set #6. Shows the number of examples for each label, as well as the number of unique signers for that sign

Label	Fall		Spring I		Spring II		All	
	Examples	Signers	Examples	Signers	Examples	Signers	Examples	Signers
alligator	109	5	63	5	166	13	338	18
bed	59	5	25	5	103	13	187	18

Table continued on next page

Table 57 – continued from previous page

Label	Fall		Spring I		Spring II		All	
	Examples	Signers	Examples	Signers	Examples	Signers	Examples	Signers
behind	47	5	20	5	95	13	162	18
black	88	5	67	5	111	12	266	17
blue	190	5	123	5	222	14	535	18
box	58	5	34	5	86	13	178	18
cat	116	5	85	5	160	13	361	18
chair	83	5	40	5	121	12	244	17
end-sentence	344	5	221	5	574	14	1139	18
flowers	101	5	67	5	131	14	299	18
green	168	5	153	5	224	13	545	18
in	121	5	71	5	156	14	348	18
on	114	5	75	5	166	13	355	18
orange	123	5	75	5	155	12	353	17
snake	122	5	58	5	146	14	326	18
spider	105	5	57	5	162	13	324	18
start-sentence	365	5	208	5	558	14	1131	18
under	151	5	77	5	206	12	434	17
wagon	36	4	20	5	64	13	120	18
wall	64	5	44	5	96	12	204	17
white	95	5	90	5	88	13	273	17

## REFERENCES

- [1] “Stanford Achievement Test, 9th Edition, Form S,” *Norms Booklet for Deaf and Hard-of-Hearing Students*, 1996. Gallaudet Research Institute.
- [2] “National Association for the Deaf: Fact sheet about American Sign Language,” 2003. <http://www.nad.org/infocenter/infotogo/asl/factsheetASL.html> (Accessed July 1, 2010).
- [3] ACHILLES, A.-C. and ORTYL, P., eds., *Citeseer: The Collection of Computer Science Bibliographies*. <http://linwww.ira.uka.de/csbib/Misc/CiteSeer/> (Accessed July 1, 2010): Universität Karlsruhe Department of Computer Science, 2010.
- [4] AIST, G. and MOSTOW, J., “Designing Spoken Tutorial Dialogue with Children to Elicit Predictable but Educationally Valuable Responses,” *Proceedings of the 10th Annual Conference of the International Speech Communication Association (Interspeech)*, September 2009.
- [5] AIST, G. and MOSTOW, J., “Predictable and Educational Spoken Dialogues: Pilot Results,” *Proceedings of the Second ISCA Workshop on Speech and Language Technology in Education (SLaTE)*, September 2009.
- [6] ALLEN, J. and HEEMAN, P. A., “README file,” in *TRAINS Dialog Corpus*, National Institute of Standards and Technology, 1993.
- [7] ANDREWS, J., VERNON, M., and LAVIGNE, M., “The deaf suspect/defendant and the bill of rights,” in *Views*, pp. 7–9, Registry of Interpreters for the Deaf, May 2006.
- [8] ANGELOVA, A., “Data Pruning,” Master’s thesis, California Institute of Technology, 2004.
- [9] ANN, J., “Contact between a Sign Language and a Written Language: Character Signs in Taiwan Sign Language,” *Pinky Extension and Eye Gaze: Language Use in Deaf Communities*, pp. 59–101, 1998.
- [10] ATHITSOS, V., NEIDLE, C., SCLAROFF, S., NASH, J., STEFAN, A., YUAN, Q., and THANGALI, A., “The American Sign Language Lexicon Video Dataset,” *Proceedings of the IEEE Workshop on Computer Vision and Pattern Recognition for Human Communicative Behavior Analysis*, 2008.
- [11] BAKER-SHENK, C. and COKELY, D., *American Sign Language: A Teacher’s Resource Text on Grammar and Culture (“The Green Book”)*. Washington D.C.: Gallaudet University Press, 1980.
- [12] BATTISON, R., *Lexical Borrowing in American Sign Language*. Silver Spring, MD: Linstok Press, 1978.

- [13] BAUER, B., HIENZ, H., and KRAISS, K., “Video-based continuous sign language recognition using statistical methods,” in *Proceedings of the 15th International Conference on Pattern Recognition*, vol. 2, pp. 463–466, September 2000.
- [14] BAUR, B., HEINZ, H., and KRAISS, K. F., “Video-based continuous sign language recognition using statistical methods,” in *Proceedings 15th IEEE International Conference on Pattern Recognition*, vol. 2, pp. 463–466, September 2000.
- [15] BOISEN, S. and BATES, M., “A practical methodology for the evaluation of spoken language systems,” in *Proceedings of the third conference on Applied natural language processing*, (Morristown, NJ, USA), pp. 162–169, Association for Computational Linguistics, 1992.
- [16] BRADSKI, G. and KAEHLER, A., *Learning OpenCV: Computer Vision with the OpenCV Library*. O’Reilly Media, 1st ed., 2008.
- [17] BRASHEAR, H., STARNER, T., LUKOWICZ, P., and JUNKER, H., “Using Multiple Sensors for Mobile Sign Language Recognition,” in *Proceedings of the Seventh IEEE International Symposium on Wearable Computers*, pp. 45–52, 2003.
- [18] BRASHEAR, H., PARK, K.-H., LEE, S., HENDERSON, V., HAMILTON, H., and STARNER, T., “American Sign Language Recognition in Game Development for Deaf Children,” in *Assets ’06: Proceedings of the 8th International ACM SIGACCESS Conference on Computers and Accessibility*, (New York, NY USA), ACM Press, 2006.
- [19] BRASHEAR, H., ZAFRULLA, Z., STARNER, T., HAMILTON, H., PRESTI, P., and LEE, S., “CopyCat: A Corpus for Verifying American Sign Language During Game Play by Deaf Children,” in *4th Workshop on the Representation and Processing of Sign Languages: Corpora and Sign Language Technologies*, Proceedings of the 7th annual international Language Resources and Evaluation Conference, (Valetta, Malta), 2010.
- [20] BURNS, S. E., “Irish Sign Language: Ireland’s Second Minority Language,” *Pinky Extension and Eye Gaze: Language Use in Deaf Communities*, pp. 233–273, 1998.
- [21] CAMPBELL, L., BECKER, D., AZARBAYEJANI, A., BOBICK, A., and PENTLAND, A., “Invariant features for 3-d gesture recognition,” in *Second International Conference on Face and Gesture Recognition*, pp. 157–162, 1996.
- [22] COOPER and BOWDEN, “Sign Language Recognition: Working with Limited Corpora,” in *Universal Access in Human-Computer Interaction. Applications and Services 5th International Conference, UAHCI 2009, Held as Part of HCI International 2009*, pp. 472–481, 2009.
- [23] COOPER, H. and BOWDEN, R., “Learning signs from subtitles: A weakly supervised approach to sign language recognition,” *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 0, pp. 2568–2574, 2009.
- [24] COOPER, H. M. and BOWDEN, R., “Sign Language Recognition Using Boosted Volumetric Features,” In *Proceedings IAPR Conference on Machine Vision Applications*, pp. 359–362, May 2007.

- [25] DE LA HAMETTE, P., LUKOWICZ, P., TRÖSTER, G., and SVOBODA, T., “Finger-mouse: A wearable hand tracking system,” in *Fourth International Conference on Ubiquitous Computing*, pp. 15–16, September 2002.
- [26] DICTA-SIGN CONSORTIUM, “DICTA-SIGN Annual Public Report 2009: Sign Language Recognition, Generation and Modelling with Applications in Deaf Communications,” March 2010.
- [27] DIVELY, V. L., “Conversational Repairs in ASL,” in *Pinky extension and eye gaze: Language use in Deaf communities* (LUCAS, C., ed.), (Washington DC), pp. 137–169, Gallaudet University Press, 1998.
- [28] DIX, A., FINLAY, J., ABOWD, G., and BEALE, R., *Human-Computer Interaction*, ch. 6.4 Iterative Design and Prototyping. Prentice Hall, 2004.
- [29] DREUW, P., NEIDLE, C., ATHITSOS, V., SCLAROFF, S., and NEY, H., “Benchmark Databases for Video-Based Automatic Sign Language Recognition,” *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC)*, 2008.
- [30] DREUW, P., NEY, H., MARTINEZ, G., CRASBORN, O., PIATER, J., MOYA, J. M., and WHEATLEY, M., “The SignSpeak Project - Bridging the Gap Between Signers and Speakers,” *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC)*, 2010.
- [31] DUDA, R., HART, P., and STORK, D., *Pattern Classification: Second Edition*. New York, New York: John Wiley and Sons, Inc., 2001.
- [32] EASTERBROOKS, S. and HUSTON, S., “The Signed Reading Fluency of Students Who are Deaf/Hard of Hearing,” *International Journal of Deaf Studies and Deaf Education*, vol. 13, no. 1, pp. 37–54, 2008.
- [33] EINFELDT, C., “Video Editing Made Easy with Kino!,” *Linux Today*, March 2009.
- [34] EKLUND, R., *Disfluency in Swedish Human–Human and Human–Machine Travel Booking Dialogues: Dissertation No. 882*. PhD thesis, Linköping Studies in Science and Technology, Linköping University, Sweden, 2004.
- [35] EMMOREY, K., *Language, Cognition, and the Brain: Insights From Sign Language Research*. Mahwah, New Jersey: Lawrence Erlbaum Associates, 2002.
- [36] ENG-JON, O. and BOWDEN, R., “A boosted classifier tree for hand shape detection,” in *Proceedings of the Sixth IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 889–894, May 2004.
- [37] FANG, G., GAO, W., and ZHAO, D., “Large Vocabulary Sign Language Recognition Based on Hierarchical Decision Trees,” in *International Conference on Multimodal Interfaces*, pp. 125–131, 2003.
- [38] FANG, G., GAO, W., and ZHAO, D., “Large-Vocabulary Continuous Sign Language Recognition Based on Transition-Movement Models,” *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, vol. 37, pp. 1–9, January 2007.

- [39] FANG, G., GAO, W., CHEN, X., WANG, C., and MA, J., “Signer-independent continuous sign language recognition based on SRN/HMM,” in *Gesture Workshop*, pp. 76–85, 2001.
- [40] FANG, G., GAO, W., and ZHAO, D., “Large vocabulary sign language recognition based on hierarchical decision trees,” in *ICMI '03: Proceedings of the 5th international conference on Multimodal interfaces*, (New York, NY, USA), pp. 125–131, ACM Press, 2003.
- [41] FLODIN, M., *Signing Illustrated: The Complete Learning Guide*. New York, New York: Penguin Group, 2004.
- [42] FURNISS, M., “Motion Capture,” *MIT Communications Forum*, December 1999.
- [43] GALES, M. and YOUNG, S., “The application of hidden markov models in speech recognition,” *Found. Trends Signal Process.*, vol. 1, no. 3, pp. 195–304, 2008.
- [44] GAO, W., FANG, G., ZHAO, D., and CHEN, Y., “Transition Movement Models for Large Vocabulary Continuous Sign Language Recognition (CSL),” in *Sixth IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 553–558, 2004.
- [45] GAO, W., MA, J., WU, J., and WANG, C., “Sign Language Recognition Based on HMM/ANN/DP,” *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 14, no. 5, pp. 587–602, 2000.
- [46] GAROFOLO, J. S., LAMEL, L. F., FISHER, W. M., FISCUS, J. G., PALLETT, D. S., DAHLGREN, N. L., and ZUE, V., “README file,” in *The DARPA TIMIT Acoustic-Phonetic Continuous Speech Corpus*, National Institute of Standards and Technology, 1986.
- [47] GODFREY, J. J. and HOLLIMAN, E., “README file,” in *Switchboard-1 Transcripts*, 1991.
- [48] GOLDIN-MEADOW, S., “What children contribute to language-learning,” *Science Progress*, vol. 82, no. 1, pp. 89–102, 1999.
- [49] GOLDIN-MEADOW, S. and MYLANDER, C., “The role of parental input in the development of a morphological system,” *Journal of Child Language*, vol. 17, pp. 527–563, October 1990.
- [50] GOLDSTEIN, S., VOGLER, C., and METAXAS, D., “Directed Acyclic Graphs Representation of Deformable Models,” *Proceedings of the IEEE Computer Society Workshop on Motion and Video Computing*, 2002.
- [51] GORDON, R., ed., *Ethnologue: Language of the World, 15th edition*. SIL International, 2008.
- [52] GROBEL, K. and ASSAN, M., “Isolated sign language recognition using hidden Markov models,” in *IEEE International Conference on Systems, Man, and Cybernetics: Computational Cybernetics and Simulation*, vol. 1, pp. 12–15, October 1997.

- [53] GUSTASON, G. and ZAWOLKOW, E., *Signing Exact English*. Los Angeles, California: Modern Signs Press, 1993.
- [54] HAMERS, J., “Cognitive and language development of bilingual children,” in *Cultural and Language Diversity and the Deaf Experience* (PARASINIS, L., ed.), pp. 51–75, Cambridge University Press, 1998.
- [55] HAMILTON, H. and LILLO-MARTIN, D., “Imitative Production of Verbs of Movement and Location: A Comparative Study,” *Sign Language Studies*, vol. 50, pp. 29–57, 1986.
- [56] HEMPHILL, C. T., GODFREY, J. J., DODDINGTON, G. R., GAROFOLO, J., and FISCUS, J., “README file,” in *DARPA Air Travel Information System*, National Institute of Standards and Technology, June 1990.
- [57] HENDERSON, V., LEE, S., BRASHEAR, H., HAMILTON, H., STARNER, T., and HAMILTON, S., “Development of an American Sign Language Game for Deaf Children,” in *Proceedings of the 4th International Conference for Interaction Design and Children*, (Boulder, CO), 2005.
- [58] HENDERSON, V., LEE, S., BRASHEAR, H., HAMILTON, H., STARNER, T., and HAMILTON, S., “Development of an American Sign Language Game for Deaf Children,” in *IDC '05: Proceeding of the 2005 Conference on Interaction Design and Children*, (New York, NY, USA), ACM Press, June 2005.
- [59] HERNANDEZ-REBOLLAR, J. L., KYRIAKOPOULOS, N., and LINDEMAN, R. W., “A New Instrumented Approach for Translating American Sign Language into Sound and Text,” in *Proceedings of the Sixth IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 547–552, 2004.
- [60] HIENZ, H. and GROBEL, K., “Automatic estimation of body regions from video images,” in *Proceedings of the International Gesture Workshop on Gesture and Sign Language in Human-Computer Interaction*, vol. 1371, pp. 135–145, London, UK: Springer-Verlag, 1998. Lecture Notes in Computer Science.
- [61] HIROHIKO SAGAWA, M. T., “A Method for Analyzing Spatial Relationships Between Words in Sign Language Recognition,” in *Proceedings of the International Gesture Workshop on Gesture-Based Communication in Human-Computer Interaction*, pp. 197–209, Springer-Verlag, January 1999. Lecture Notes on Computer Science.
- [62] HOLDEN, E.-J. and OWENS, R., “Visual sign language recognition,” *Lecture Notes in Computer Science*, vol. 2032, 2001.
- [63] HOLT, J., TRAXLER, C. B., and ALLEN, T. E., “Interpreting the Scores: A User’s Guide to the 9th Edition Stanford Achievement Test for Educators of Deaf and Hard-of-Hearing Students,” *Gallaudet Research Institute Technical Report*, vol. 91, no. 1, 1997. Gallaudet Research Institute.
- [64] IMAGAWA, K., MATSUO, H., ICHIRO TANIGUCHI, R., ARITA, D., LU, S., and IGI, S., “Recognition of local features for camera-based sign language recognition system,” *Pattern Recognition, International Conference on*, vol. 4, p. 4849, 2000.

- [65] JOHNSON, R., LIDDELL, S., and ERTING, C., “Unlocking the curriculum,” GRI Working Paper 89(3), Gallaudet Research Institute, Washington, DC, 1989.
- [66] JONES, K. S. and GALLIERS, J. R., *Evaluating natural language processing systems: An analysis and review*. Lecture Notes in Artificial Intelligence, Springer, 1995.
- [67] JUNG HOLDEN, E., LEE, G., and OWENS, R., “Automatic recognition of colloquial australian sign language,” in *In WACV-MOTION 05: Proceedings of the IEEE Workshop on Motion and Video Computing (WACV/MOTION05) - Volume 2*, pp. 183–188, IEEE Computer Society, 2005.
- [68] JURAFSKY, D. and MARTIN, J. H., *Speech and Language Processing*. Upper Saddle River, New Jersey: Prentice Hall, 2000.
- [69] KADOUS, W., “Recognition of Australian Sign Language using instrumented gloves,” Master’s thesis, University of New South Wales, October 1995.
- [70] KANNAPELL, B., “An examination of deaf college students’ attitudes toward ASL and English,” in *The Sociolinguistics of the Deaf Community* (LUCAS, C., ed.), pp. 191–210, Gallaudet University Press, 1989.
- [71] KIM, J. S., JANG, W., and BIEN, Z., “A Dynamic Gesture Recognition System for the Korean Sign Language KSL,” *IEEE Transactions on Systems, Man and Cybernetics*, vol. 26, no. 2, pp. 354–359, 1996.
- [72] KING, M., “Evaluating natural language processing systems,” *Communications of the ACM*, vol. 39, no. 1, pp. 73–79, 1996.
- [73] KLIMA, E. and BELLUGI, U., *The Signs of Language*. Cambridge, MA: Harvard University Press, 1979.
- [74] KOBAYASHI, T. and HARUYAMA, S., “Partly-hidden markov model and its application to gesture recognition,” in *Acoustics, Speech, and Signal Processing, 1997. ICASSP-97., 1997 IEEE International Conference on*, vol. 4, pp. 3081–3084 vol.4, 21-24 1997.
- [75] KRASHEN, S., “Adult second language acquisition and learning: A review of theory and practice,” in *Second Language Acquisition and Foreign Language Teaching* (GINGRAS, R., ed.), Washington, DC: Center for Applied Linguistics, 1978.
- [76] KRASHEN, S., “The Theoretical and Practical Relevance of Simple Codes in Second Language Acquisition,” in *Research in Second Language Acquisition: Selected Papers of the Los Angeles Second Language Acquisition Research Forum. Issues In Second Language Research* (SCARCELLA, R. and KRASHEN, S., eds.), Rowley, MA: Newberry House, 1980.
- [77] KUBALA, F., BARRY, C., BATES, M., BOBROW, R., FUNG, P., INGRIA, R., MAKHOUL, J., NGUYEN, L., SCHWARTZ, R., and STALLARD, D., “BBN BYBLOS and HARC February 1992 ATIS benchmark results,” in *HLT ’91: Proceedings of the Workshop on Speech and Natural Language*, (Morristown, NJ, USA), pp. 72–77, Association for Computational Linguistics, 1992.

- [78] LEDERBERG, A. and SPENCER, P., “Vocabulary development of young deaf and hard of hearing children,” in *Context, Cognition, and Deafness* (CLARK, M., MARSCHARK, M., and KARCHMER, M., eds.), pp. 73–92, Washington, DC: Gallaudet University Press, 2001.
- [79] LEE, C. and XU, Y., “Online, interactive learning of gestures for human/robot interfaces,” in *IEEE International Conference on Robotics and Automation*, vol. 4, (Minneapolis, MN), pp. 2982–2987, 1996.
- [80] LEE, S., HENDERSON, V., BRASHEAR, H., STARNER, T., HAMILTON, S., and HAMILTON, H., “User-centered Development of a Gesture-based American Sign Language Game,” in *NTID Instructional Technology and Education of the Deaf Symposium*, (Rochester, NY), 2005.
- [81] LEE, S., HENDERSON, V., HAMILTON, H., STARNER, T., BRASHEAR, H., and HAMILTON, S., “A Gesture-based American Sign Language Game for Deaf Children,” in *Proceedings of CHI*, (Portland, Oregon), pp. 1589–1592, 2005.
- [82] LEONARD, R. G., “A Database for Speaker-Independent Digit Recognition,” in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, vol. 3, p. 42.11, 1984.
- [83] LEONARD, R. G. and DODDINGTON, G., “README file,” in *Studio Quality Speaker-Independent Connected-Digit Corpus (TIDIGITS)*, National Institute of Standards and Technology, 1984.
- [84] LIANG, R. and OUHYOUNG, M., “A real-time continuous gesture interface for Taiwanese Sign Language,” in *ACM Symposium on User Interface Software and Technology*, 1997.
- [85] LIANG, R. and OUHYOUNG, M., “A Real-Time Continuous Gesture Recognition System for Sign Language,” in *Third International Conference on Automatic Face and Gesture Recognition*, pp. 558–565, 1998.
- [86] LIDDELL, S. K., *Grammar, gesture and meaning in American Sign Language*. Cambridge University Press, 2003.
- [87] LIDDELL, S. K. and METZGER, M., “Gesture in sign language discourse,” *Journal of Pragmatics*, vol. 30, pp. 657–697, December 1998.
- [88] LOEDING, B. L., SARKAR, S., PARASHAR, A., and KARSHMER, A. I., “Progress in Automated Computer Recognition of Sign Language,” in *Proceedings of International Conference on Computers Helping People with Special Needs*, pp. 1079–1087, Springer-Verlag Berlin Heidelberg, 2004. Lecture Notes in Computer Science.
- [89] LYONS, K., BRASHEAR, H., WESTEYN, T., KIM, J. S., and STARNER, T., “GART: The Gesture and Activity Recognition Toolkit,” in *Human-Computer Interaction. HCI Intelligent Multimodal Interaction Environments: 12th International Conference, HCI International 2007, Beijing, China. Proceedings, Part III*, vol. 4552, London, UK: Springer-Verlag, July 2007. Lecture Notes in Computer Science.

- [90] MANDAL, M. K., ASTHANA, H. S., DWIVEDI, C. B., and BRYDEN, M. P., “Hand Preference in the Deaf,” in *Journal of Developmental and Physical Disabilities*, vol. 11, 1999.
- [91] MATIC, N., GUYON, I., BOTTOU, L., DENKER, J. S., and VAPNIK, V. N., “Computer aided cleaning of large databases for character recognition,” in *Proceedings of the 11th IAPR International Conference on Pattern Recognition, Conference B: Pattern Recognition Methodology and Systems*, vol. II, (La Hague), pp. 330–333, IEEE, September 1992.
- [92] MATSUO, H., IGI, S., LU, S., NAGASHIMA, Y., TAKATA, Y., and TESHIMA, T., “The Recognition Algorithm with Non-Contact for Japanese Sign Language Using Morphological Analysis,” *Gesture and Sign Language in Human-Computer Interaction: Proceedings of the International Gesture Workshop*, vol. 1371, 1997.
- [93] MAYBERRY, R. I. and EICHEN, E. B., “The Long-Lasting Advantage of Learning Sign Language in Childhood: Another Look at the Critical Period for Language Acquisition,” *Journal of Memory and Language*, vol. 30, pp. 486–498, 1991.
- [94] MAYBERRY, R., LOCK, E., and KAZMI, H., “Linguistic ability and early language exposure,” *Nature*, vol. 41, p. 38, May 2002.
- [95] MCGUIRE, R. M., HERNANDEZ-REBOLLAR, J., STARNER, T., HENDERSON, V., BRASHEAR, H., and ROSS, D. S., “Towards a One-Way American Sign Language Translator,” in *Proceedings of the Sixth IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 620–625, 2004.
- [96] MESSING, L., ERENSHTEYN, R., FOULDS, R., GALUSKA, S., and STERN, G., “American Sign Language computer recognition: Its present and its promise. 1994,” in *International Society for Augmentative and Alternative Communication: Conference Book and Proceedings*, (Maastricht, Netherlands), pp. 289–291, 1994.
- [97] MITCHELL, R. and KARCHMER, M., “Chasing the mythical ten percent: Parental hearing status of deaf and hard of hearing students in the united states,” *Sign Language Studies*, vol. 4, pp. 138–163, Winter 2004.
- [98] MURAKAMI, K. and TAGUCHI, H., “Gesture recognition using recurrent neural networks,” in *Proceedings of CHI*, pp. 237–241, 1991.
- [99] NEIDLE, C., MICHAEL, N., NASH, J., and METAXAS, D., “A Method for Recognition of Grammatically Significant Head Movements and Facial Expressions, Developed through use of a Linguistically Annotated Video Corpus,” in *Proceedings of the Workshop on Formal Approaches to Sign Languages, held as part of the 21st European Summer School in Logic, Language and Information*, (Bordeaux, France), July 2009.
- [100] NEIDLE, C., SCLAROFF, S., and ATHITSOS, V., “SignStream: A Tool for Linguistic and Computer Vision Research on Visual-Gestural Language Data,” *Behavior Research Methods, Instruments, and Computers*, vol. 33, no. 3, pp. 311–320, 2001.
- [101] NEWPORT, E. L., “Maturational Constraints on Language Learning,” *Cognitive Science*, vol. 14, pp. 11–28, 1990.

- [102] NICHOLSON, A., *Generalization error estimates and training data valuation*. PhD thesis, California Institute of Technology, 2002.
- [103] OHKI, M., SAGAWA, H., SAKIYAMA, T., OOHIRA, E., IKEDA, H., and FUJISAWA, H., “Pattern recognition and synthesis for sign language translation system,” in *Assets '94: Proceedings of the first annual ACM conference on Assistive technologies*, (New York, NY, USA), pp. 1–8, ACM Press, 1994.
- [104] OLIVER, N., PENTLAND, A., and BERARD, F., “Lafter: Lips and Face Real Time Tracker,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 123–129, 1997.
- [105] ONG, S. and RANGANATH, S., “Automatic sign language analysis: a survey and the future beyond lexical meaning,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 27, pp. 873 – 891, jun 2005.
- [106] PADDEN, C. and HUMPHRIES, T., *Inside Deaf Culture*. Cambridge, Massachusetts: Harvard University Press, 2005.
- [107] PADDEN, C. A., “The relation between space and grammar in asl verb morphology,” *Sign language research: Theoretical Issues*, pp. 118–132, 1990.
- [108] PAUL, P. V., *Language and Deafness*. Sudbury, MA: Jones and Bartlett Learning, fourth ed., 2008.
- [109] POTAMIANOS, A. and PERAKAKIS, M., *Human-Computer Interfaces to Multimedia Content: A Review*. Springer, 2008.
- [110] RABINER, L., “A tutorial on hidden Markov models and selected applications in speech recognition,” *Proceedings of the IEEE*, vol. 77, pp. 257–286, February 1989.
- [111] RISLEY, T. and HART, B., *Meaningful Differences in the Everyday Experience of Young American Children*. Baltimore, MD: Paul H. Brookes Publishing, 1995.
- [112] ROE, D. B. and WILPON, J. G., *Voice communication between humans and machines*. National Academies Press, 1994.
- [113] ROSS, D., “Interview,” July 2003.
- [114] SADOWSKI, W. J., “Capabilities and Limitations of Wizard of Oz Evaluations of Speech User Interfaces,” 2001.
- [115] SAGAWA, H. and TAKEUCHI, M., “A method for recognizing a sequence of sign language words represented in a Japanese Sign Language sentence,” in *Proceedings of the Fourth IEEE International Conference on Automatic Face and Gesture Recognition*, (Grenoble, France), pp. 434–439, March 2000.
- [116] SAGAWA, H. and TAKEUCHI, M., “Development of an information kiosk with a sign language recognition system,” in *CUU '00: Proceedings on the 2000 conference on Universal Usability*, (New York, NY, USA), pp. 149–150, ACM Press, 2000.

- [117] SAGAWA, H. and TAKEUCHI, M., “A teaching system of japanese sign language using sign language recognition and generation,” in *MULTIMEDIA '02: Proceedings of the tenth ACM international conference on Multimedia*, (New York, NY, USA), pp. 137–145, ACM Press, 2002.
- [118] SAGAWA, H., TAKEUCHI, M., and OHKI, M., “Description and recognition methods for sign language based on gesture components,” in *IUI '97: Proceedings of the 2nd international conference on Intelligent user interfaces*, (New York, NY, USA), pp. 97–104, ACM Press, 1997.
- [119] SANDLER, W., “Sign Language Phonology,” *The Oxford International Encyclopedia of Linguistics*, 2003.
- [120] SANDLER, W., “An Overview of Sign Language Linguistics,” *Encyclopedia of Language and Linguistics*, vol. 11, pp. 328–338, 2005.
- [121] SANDLER, W. and LILLO-MARTIN, D. C., *Sign Language and Linguistic Universals*. Cambridge, UK: Cambridge University Press, 2006.
- [122] SCHEGLOFF, E. A., JEFFERSON, G., and SACKS, H., “The preference for self-correction in the organization of repair in conversation,” *Language*, vol. 53, pp. 361–382, 1977.
- [123] SCHIEFULBUSCH, R. L. and BRICKER, D. D., *Early Language: Acquisition and Intervention*. Baltimore: University Park Press, 1981.
- [124] SCHLENZIG, J., HUNTER, E., and JAIN, R., “Recursive identification of gesture inputs using hidden Markov models,” *Proceedings Second Annual Conference on Applications of Computer Vision*, pp. 187–194, December 1994.
- [125] SIDNELL, J., *Conversational Analysis: An Introduction*. West Sussex, UK: John Wiley and Sons, 2010.
- [126] SIGAL, L., SCLAROFF, S., and ATHITSOS, V., “Skin Color-Based Video Segmentation under Time-Varying Illumination,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 7, pp. 862–877, 2004.
- [127] SPENCER, P., “Communication behaviors of infants with hearing loss and their hearing mothers,” *Journal of Speech and Hearing Research*, vol. 36, pp. 311–321, 1993.
- [128] SPENCER, P., “The expressive communication of hearing mothers and deaf infants,” *American Annals of the Deaf*, vol. 138, pp. 275–283, 1993.
- [129] SPENCER, P. and LEDERBERG, A., “Different Modes, Different Models: Communication and Language of Young Deaf Children and Their Mothers,” in *Communication and Language: Discoveries from Atypical Development* (ROMSKI, M., ed.), pp. 203–230, Harvard University Press, 1997.
- [130] STARNER, T. and PENTLAND, A., “Visual Recognition of American Sign Language Using Hidden Markov Models,” in *Proceedings of the International Workshop on Automatic Face and Gesture Recognition*, 1995.

- [131] STARNER, T., WEAVER, J., and PENTLAND, A., “Real-Time American Sign Language Recognition Using Desk and Wearable Computer Based Video,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 12, pp. 1371–1375, 1998.
- [132] STOKOE, W., “Sign Language Structure: An Outline of the Visual Communication Systems of the American Deaf,” *Studies in Linguistics, Occasional Papers 8*, 1960.
- [133] STORRING, M., ANDERSEN, H., and GRANUM, E., “Skin Colour Detection under Changing Lighting Conditions,” in *Proceedings of the Seventh Symposium on Intelligent Robotics Systems*, pp. 187–195, 1999.
- [134] STORRING, M., ANDERSEN, H., and GRANUM, E., “Estimation of the Illuminant Colour from Human Skin Colour,” in *Proceedings of International Conference on Automatic Face and Gesture Recognition*, pp. 64–69, 2000.
- [135] TAKAHASHI, T. and KISHINO, F., “Hand gesture coding based on experiments using a hand gesture interface device,” *SIGCHI Bulletin*, vol. 23, no. 2, pp. 67–73, 1991.
- [136] TEN HOLT, G., “The Eye of the Beholder: Automatic Recognition of Dutch Sign Language,” Master’s thesis, University of Groningen, Netherlands, 2004.
- [137] TEN HOLT, G., HENDRIKS, P., and ANDRINGA, T., “Why Don’t You See What I Mean? Prospects and Limitations of Current Automatic Sign Recognition Research,” *Sign Language Studies*, vol. 6, Summer 2006.
- [138] VALLI, C. and LUCAS, C., *Linguistics of American Sign Language: An Introduction. First Edition*. Washington DC: Galludet University Press, 1998.
- [139] VALLI, C. and LUCAS, C., *Linguistics of American Sign Language: An Introduction. Third Edition*. Washington DC: Galludet University Press, 2010.
- [140] VIOLA, P. and JONES, M., “Robust Real-time Object Detection,” *International Journal of Computer Vision*, p. 137154, 2001.
- [141] VOGLER, C. and METAXAS, D., “Adapting Hidden Markov Models for ASL Recognition by Using Three-Dimensional Computer Vision Methods,” in *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, pp. 156–161, 1997.
- [142] VOGLER, C. and METAXAS, D., “ASL Recognition Based on a Coupling Between HMMs and 3D Motion Analysis,” in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 363–369, 1998.
- [143] VOGLER, C. and METAXAS, D., “Parallel hidden Markov models for American sign language recognition,” in *Proceedings of the Seventh IEEE International Conference on Computer Vision*, vol. 1, pp. 116–122, 1999.
- [144] VOGLER, C., SUN, H., and METAXAS, D., “A framework for motion recognition with applications to American sign language and gait recognition,” in *Proceedings of Workshop on Human Motion*, pp. 33–38, 2000.

- [145] VOGLER, C. and METAXAS, D., “Adapting Hidden Markov Models for ASL recognition by using three-dimensional computer vision methods,” in *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, (Orlando, FL), pp. 156–161, October 1997.
- [146] VOGLER, C. and METAXAS, D., “Handshapes and Movements: Multiple-Channel American Sign Language Recognition,” in *Gesture-Based Communication in Human-Computer Interaction*, vol. 2915, pp. 247–258, Springer-Verlag, January 2004. Lecture notes in Artificial Intelligence.
- [147] WANG, C., GAO, W., and MA, J., “A real-time large vocabulary recognition system for chinese sign language,” in *Gesture Workshop*, pp. 86–95, 2001.
- [148] WATERWORTH, J. A., “Conversational analysis and human-computer dialog design,” *SIGCHI Bull.*, vol. 18, no. 2, pp. 54–55, 1986.
- [149] WEAVER, K. A., HAMILTON, H., ZAFRULLA, Z., BRASHEAR, H., STARNER, T., PRESTI, P., and BRUCKMAN, A., “Improving the Language Ability of Deaf Signing Children through an Interactive American Sign Language-Based Video Game,” in *Proceedings of 9th International Conference of the Learning Sciences*, June 2010.
- [150] WESTEYN, T., PRESTI, P., and STARNER, T., “A Naive Technique for Correcting Time-Series Data for Recognition Applications,” in *Proceedings of the Thirteenth IEEE International Symposium on Wearable Computers (ISWC 2009)*, IEEE Computer Society, Sept 04-07 2009.
- [151] WESTEYN, T., BRASHEAR, H., ATRASH, A., and STARNER, T., “Georgia Tech Gesture Toolkit: Supporting Experiments in Gesture Recognition,” in *ICMI '03: Proceedings of the 5th International Conference on Multimodal Interfaces*, (New York, NY, USA), ACM Press, 2003.
- [152] XU, M., RAYTCHEV, B., SAKAUE, K., HASEGAWA, O., KOIZUMI, A., TAKEUCHI, M., and SAGAWA, H., “A Vision-Based Method for Recognizing Non-manual Information in Japanese Sign Language,” in *ICMI '00: Proceedings of the Third International Conference on Advances in Multimodal Interfaces*, vol. 1948, pp. 572–581, London, UK: Springer-Verlag, 2000.
- [153] YAMATO, J., OHYA, J., and ISHII, K., “Recognizing Human Action in Time-Sequential Images using Hidden Markov Models,” in *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 379–385, 1992.
- [154] YAMATO, J., OHYA, J., and ISHII, K., “Recognizing human action in time-sequential images using hidden Markov models,” *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 379–385, 1992.
- [155] YANG, J., WEIER, L., and WAIBEL, A., “Skin-Color Modeling and Adaptation,” in *Proceedings of Asian Conference on Computer Vision*, pp. 687–694, 1998.
- [156] YANG, M. H., AHUJA, N., and TABB, M., “Extraction of 2d motion trajectories and its application to hand gesture recognition,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 8, pp. 1061–1074, 2002.

- [157] YANG, R., SARKAR, S., and LOEDING, B., “Enhanced level building algorithm for the movement epenthesis problem in sign language recognition,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, 2007.
- [158] YANG, R., SARKAR, S., and LOEDING, B., “Handling movement epenthesis and hand segmentation ambiguities in continuous sign language recognition using nested dynamic programming,” *The IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, pp. 462–477, March 2010.
- [159] YANG, R., SARKAR, S., LOEDING, B., and KARSHMER, A., “Efficient Generation of Large Amounts of Training Data for Sign Language Recognition: A Semi-Automatic Tool,” *Proceedings of International Conference on Computers Helping People*, 2006.
- [160] YIN, P., *Segmental Discriminative Analysis for American Sign Language Recognition and Verification*. PhD thesis, Georgia Institute of Technology, College of Computing, Atlanta, GA, 2010.
- [161] YOUNG, S., EVERMANN, G., GALES, M., HAIN, T., KERSHAW, D., LIU, Z., MOORE, G., ODELL, J., OLLASON, D., POVEY, D., VALTCHEV, V., and WOODLAND, P., “The HTK Book (for HTK Version 3.4),” 2009.
- [162] YULE, G., *The Study of Language 4th Edition*. Cambridge, UK: Cambridge University Press, 2010.
- [163] ZAFRULLA, Z., BRASHEAR, H., HAMILTON, H., and STARNER, T., “A novel approach to American Sign Language (ASL) Phrase Verification using Reversed Signing,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2010.
- [164] ZAFRULLA, Z., BRASHEAR, H., HAMILTON, H., and STARNER, T., “Towards an American Sign Language Verifier for Educational Game for Deaf Children,” in *Proceedings of International Conference on Pattern Recognition*, 2010.
- [165] ZIEREN, J. and KRAISS, K.-F., “Non-Intrusive Sign Language Recognition for Human-Computer Interaction,” *Proceedings of the 9th IFAC/IFIP/IFORS/IEA Symposium on Analysis Design, and Evaluation of Human-Machine Systems*, September 2004.
- [166] ZUE, V., GLASS, J., GODDEAU, D., GOODINE, D., HIRSCHMAN, L., PHILLIPS, M., POLIFRONI, J., and SENEFF, S., “The MIT ATIS system: February 1992 progress report,” in *HLT '91: Proceedings of the Workshop on Speech and Natural Language*, (Morristown, NJ, USA), pp. 84–88, Association for Computational Linguistics, 1992.
- [167] ZUE, V. W. and GLASS, J. R., “Conversational Interfaces: Advances and Challenges,” *Proceedings of the IEEE, Special Issue on Spoken Language Processing*, vol. 88, pp. 1166–1180, August 2000.