Corpus-Based Training of Action-Specific Language Models

Lars Schillingmann* and **Sven Wachsmuth** and **Britta Wrede**

Bielefeld University, 33615 Bielefeld, Germany,

{lschilli, swachsmu, bwrede}@techfak.uni-bielefeld.de

http://www.techfak.uni-bielefeld.de/ags/ai/

Abstract

Especially in noisy environments like in human-robot interaction, visual information provides a strong cue facilitating a robust understanding of speech. In this paper, we consider the dynamic visual context of actions perceived by a camera. Based on an annotated multi-modal corpus of people who verbally explain tasks while they perform them, we present an automatic strategy for learning action-specific language models. The approach explicitly deals with the asynchrony of actions and verbal descriptions and includes an automatic parameter optimization based on a perplexity measure. Results show that a significant improvement of the word accuracy can be achieved using a dynamic switching of action-specific language models.

1 Introduction

While speech recognition is an easy task for humans even under difficult acoustic conditions, current ASR systems still cannot compete with humans (Potamianos et al., 2003). This is especially true in human-robot interaction, where one has to deal with spontaneous speech effects, noisy environments, communicative gestures, and a frequent referencing to visual objects and events. In this case, speech recognition and understanding becomes a multi-modal issue. This has also been emphasized by several psychological studies that suggest a very early interaction between vision and speech processing (Spivey et al., 2001). For the practical development of speech understanding components for robotic interfaces, there are three implications. First, there is a need for multi-modal corpora in order to train and evaluate more sophisticated speech recognition models. Secondly, visual and acoustic speech events need to be synchronized and aligned with regard to semantic content for learning as well as interpretation. Thirdly, new strategies for the early integration of visual information into the speech recognition process need to be developed. In this paper, we focus on the first and second issues and show first results for the third.

The integration of speech and visual context can be treated on different levels of processing that depend on the kind visual information considered. Motivated by the McGurk effect (1976) audiovisual speech recognition (AVSR) systems have been developed. These systems integrate acoustic features with those extracted from the speakers face. This is an approximately synchronous process during speech production. In AVSR, typically Hidden Markov Models (HMMs) are used for modelling the acoustic and visual features. The approaches mostly differ in the handling of slight asynchrony between the two feature streams. The methods range from simple feature concatenation which does not allow asynchrony at all up to more flexible HMM architectures (e.g. Product-HMMs) allowing ca. 100 ms of asynchrony in practice (Potamianos et al., 2003).

Other systems proposed integrate features from a static visual scene into speech recognition. Knowledge inferred from a visual scene can be used to generate grammars for object descriptions (Naeve et al., 1995). These grammars are used as language model to improve speech recognition. Deb Roy (2005) reports a system, which fuses knowledge of the visual semantics of language and the specific contents of a visual scene during speech processing. Based on

*Partially supported by the Federal Ministry of Education and Research Germany (Joint Project DESIRE)

Proceedings of the 8th SIGdial Workshop on Discourse and Dialogue, pages 95–102,
Antwerp, September 2007.

the current scene layout the system generates possible word sequences for object descriptions from a probabilistic grammar. These are weighted by a likelihood associated with each object in the scene. The result is a bi-gram model, which is dynamically updated using a visual attention mechanism incorporating the partially processed utterance. This model is used to bias speech recognition. Both approaches have in common that the scene information remains static during speech processing. Thus, the synchronization problem can be neglected and the integration is done on the level of utterances. In this case also late integration schemes are possible that infer a joint multi-modal meaning after a word sequence has been recognized (Wachsmuth and Sagerer, 2002).

The timing and synchronization becomes relevant when dynamic visual events are considered as visual context. Two different cases can be distinguished. On the one hand, communicative gestures like pointing provide information that is directly related to the syntactic structure of the sentence. As a consequence, these are approximately synchronized with the corresponding noun phrases and partially marked in the wording. In this area, different research groups have started to collect multimodal corpora (Green et al., 2006; Wolf and Bugmann, 2005; Maas and Wrede, 2006). However, in these settings, the scene environment is still static and the kind of visual information provided is of limited use in speech recognition.

On the other hand, human actions or action sequences that are verbally commented are the most informative but also most flexible case. Usable corpora for speech recognition training as well as evaluation are still rare. Integrating this information into speech recognition broaches two problems. First, humans do not execute actions synchronously while describing a task verbally. The degree of asynchrony lays in a range of several seconds as reported in (Wolf and Bugmann, 2006). Hence, it is not possible to integrate this information using HMM architectures as used in AVSR. Second, the actions change in the course of an utterance. Thus, the contextual information is not static as in the previous systems utilizing visual scene contents.

In this paper, we present a corpus-based method for training and optimising action-specific language models. The goal is to improve recognition accuracy by using these models during speech processing. Training data for the language models is collected using a scenario described in section 2. Section 3 describes our method of associating utterance parts to actions. The resulting action-specific training data is used in an automated language model training and optimisation process. The results of this process are discussed in section 4.

2 Scenario and data collection

Figure 1: A test subject describes a task while performing it.

Our scenario resembles a situation in which a user teaches a new task to a robotic system. A test subject sits in front of a table with several objects (e.g. a cup and a plant) on it that can be utilized for different manipulative actions (Figure 1). Only a subset of the objects is relevant for the following demonstration. The subject is instructed to explain some simple tasks to the system while performing the corresponding action sequence. In order to suppress deictic gestures and too complex descriptions they have to imagine, that their communication partner is intelligent and knows the setup. The tasks are watering a plant, preparing tea and preparing coffee. In order to generate more varying utterances the test subjects have to perform each task twice with three different object layouts. The second time they are additionally instructed to name colours and object relationships if possible. The utterances are recorded using a headset microphone and the scene is recorded by video. A corpus is collected containing the utterance transcriptions and time intervals, which annotate the actions. The actions performed are annotated in the video based on an abstraction hierarchy

as depicted in Figure 2). The choice of the compositional granularity was based on two reasons. First, the corresponding primitives can be detected using a pre-trained trajectory based action recogniser (Li et al., 2006). Secondly, the verbalization happened on that level due to the instructions given.

The resulting corpus consists of 195 utterances from 11 test subjects (17.7 utterances per person). The overall length is about 38 minutes. The average utterance length is about 12.7 s with about 33 words per utterance. The entire corpus includes 6 429 words with a lexicon size of 288 different words. The videos are annotated with 11 different actions. The average length of an action interval is 1.75 s. All in all 999 intervals with an overall length of about 29 minutes have been annotated. Each utterance contains 5.5 actions in average.

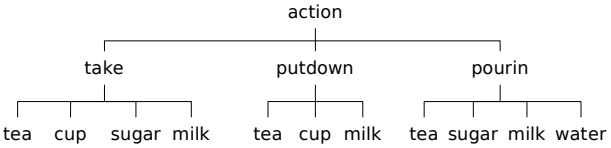

Figure 2: Hierarchic structure of actions used for annotation.

The following section describes how action-specific language models are created using this corpus.

3 Action-Specific Language Models

Speech recognition models are typically formulated distinguishing acoustic and language models. The standard technique for language models are n-grams that have proven their effectiveness over many years (Rosenfeld, 2000). For acquiring realistic language models, n-grams need to be trained using a representative sample. In the present approach, we assume that the wording will be biased by the action, which the speaker performs and describes in parallel. Thus, we aim at the estimation of action-specific language models. In order to gain corresponding action-specific samples two problems need to be solved. First, a method is required, which is able to associate speech with action intervals in order to extract action-specific parts from an utterance. Secondly, our approach requires temporal information (word intervals) for both the actions and the speech. The utterance transcriptions from the above-described corpus are not annotated with temporal information in contrast to the video annotation. Manual annotation on that level of detail is expensive. Thus, we use an automated approach, which is described in the next section. Afterwards we elaborate on our approach to the first problem.

3.1 Gaining Time Information

The temporal information of an utterance with a known transcription can be gained by using a so-called forced alignment. Our speech recogniser (Fink, 1999) uses Hidden Markov Models (HMMs) as acoustic models. Existing models trained on a speech corpus are used. Words not in the lexicon are defined by new compound models based on phoneme HMMs. In a forced alignment, the model topology is restricted in accordance with each utterance transcription. This means the order of word models is fixed for each transcription ensuring a correct alignment although the acoustic quality varies depending on the speaker. Since the transcription does not contain pauses or spontaneous speech effects, the model topology needs to be adapted accordingly. An "<other>" model for these effects is optionally allowed between words. Figure 3 shows a schematic diagram of the model topology. For

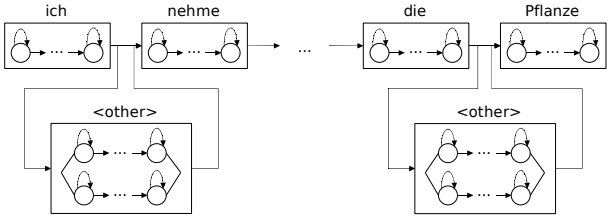

Figure 3: Schematic diagram of a HMM topology with fixed word model order and optional "<other>" models.

each utterance, a sequence of MFCC feature vectors is extracted following standard speech recognition techniques. The Viterbi algorithm is used to calculate the state sequence $\boldsymbol{s}$ through the model topology which produces the feature vector sequence $\boldsymbol{o}$ with the maximum probability given the HMM λ:

$$\boldsymbol{s}^* = \underset{\boldsymbol{s}}{\operatorname{argmax}}\, P(\boldsymbol{o}, \boldsymbol{s}|\lambda) \qquad (1)$$

After the Viterbi alignment, the resulting state sequence can be used to calculate the time interval for

each word since the frame length used during feature extraction is known. After this step, the temporal information is available for both the utterance transcription and the action annotation. The following section explains the next step where the temporal information is used to associate utterance parts with actions.

3.2 Pairing of Speech and Actions

The main problem when speech has to be associated with action intervals is that the utterance parts semantically belonging to actions are asynchronous on the time-line (Wolf and Bugmann, 2006). Thus, a distance measure $d(w_i, a_j)$ is calculated between each word w_i and action a_j. A set of tolerance parameters is used to decide if a word is assigned to an action. By choosing these parameters appropriately, the asynchrony between speech and actions can be respected. Since the time shift is not longer than several seconds this procedure is suitable. Multiple cases have to be handled when calculating with temporal intervals, which are systematically structured by Allen's calculus (Allen, 1983). Our method uses a subset of these relationships. Each type of action uses independent tolerance parameters to the left h_j^l and the right h_j^r. They are used depending if w_i is before or after a_j respectively. Pauses detected during the forced alignment give hints about the change of an action. Thus, silence is weighted additionally using a penalty parameter g_j so that silence between an action and a word further increases the temporal difference. Figure 4 illustrates the distance measure when silence has to be considered.

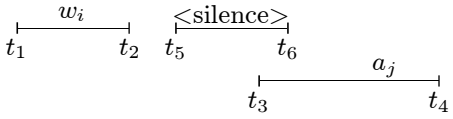

Figure 4: The distance function between two word intervals under the above constellation is defined as $d(w_i, a_j) = t_3 - t_2 + g_j \cdot (t_3 - t_5)$.

A word is associated with an action if the following condition is true:

$$-h_j^r < d(w_i, a_j) < h_j^l \qquad (2)$$

Figure 5 gives a simple example about the assignment strategy. The tolerance parameters are determined automatically and individually for each language model using an optimisation method, which is described in section 3.4.

3.3 Language Model Training

The objective of the language model training is to create a n-gram-model for each action type, which predicts the action-specific utterance parts most accurately. These models could directly be trained with the results of the above assignment strategy but it is likely that these models become too specific. Therefore, the training data is structured using the hierarchy defined in figure 2. The top level refers to the complete utterance. The second level addresses utterance parts on a more general action level e.g. "take" or "put". The third level reaches the highest level of granularity with action-object specific utterance parts. During training each level can be weighted using an individual factor (see figure 6). The set of weighting factors is specific for each language model. Thus, each language model has an individual degree of specialisation depending on these factors. The training data required in this process is generated using the speech and action pairing process with an individual parameter set. Both the pairing parameters and the weighting factors are optimised specifically for each language model using a method described in the following section.

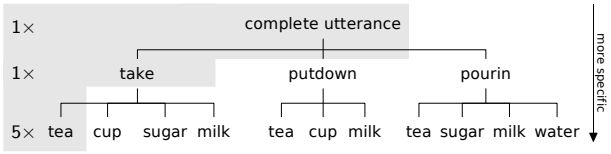

Figure 6: Structure of the training data using the action hierarchy. The highlighted path shows by example, which parts are used and weighted to train one language model.

During model estimation, absolute discounting and backing-off are used to handle unseen events. The counts $c(\boldsymbol{y}z)$ of a word z with history $\boldsymbol{y}$ are modified with an absolute value β in order to gain probability mass for unseen events so that the relative frequencies are defined as:

$$f^*(z|\boldsymbol{y}) = \frac{c(\boldsymbol{y}z) - \beta}{c(\boldsymbol{y}\cdot)} \qquad \forall \boldsymbol{y}z \; c(\boldsymbol{y}z) > \beta \qquad (3)$$

Where $c(\boldsymbol{y}\cdot)$ denotes all events with history $\boldsymbol{y}$.

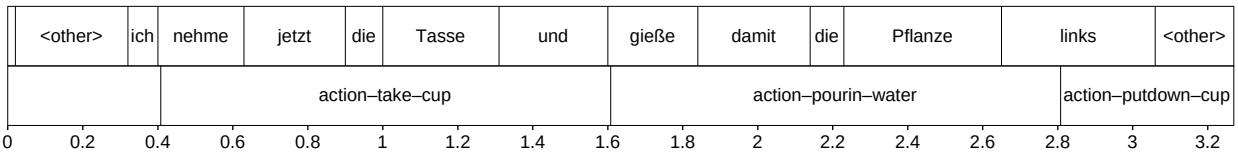

Figure 5: Augmented utterance transcription and action annotation on one time axis ($t[s]$). Assuming pourin-water has a tolerance of 0.5 s to the left and 0 s to the right the part "Tasse und gieße damit die Pflanze links" is assigned to this action.

3.4 Parameter Optimisation

In the above sections, we have introduced several parameters. The tolerance parameters and the penalty factors for silence sum up to 33 in total considering all 11 action types. In addition, the weighting factors in the training data structure count 33 in total. This large number of free parameters cannot efficiently be determined manually. Thus, we use an optimisation method, which uses the perplexity to measure the quality of the action-specific models. We firstly describe the method in general and go into detail in the next paragraph.

In order to compute the perplexity a test sample is required. Since our corpus is relatively small, the choice of the test sample has large influence on the perplexity. Therefore the perplexity is computed using a leave-one-out cross validation (Kohavi, 1995). The utterances of one person are used as testing data on each run; the others are used for training. Firstly, a parameter set with the above parameters is generated. This parameter set is used to train language models with the method described in the last two sections. The testing data is gained using the same parameter set. Secondly, the perplexity is computed for each excluded test subject. The average perplexity regarding an action-specific language model is the final measurement of this model and the underlying parameter set. Thus, a parameter optimisation also finds the tolerance parameters for speech action assignment. The asynchrony between speech and actions is respected this way. This method depends on the assumption that actions frame semantic units, which are verbalised similarly. Therefore, a correct assignment of speech to actions results in a better perplexity rating.

In detail, the optimisation is realised by evaluating a large number of parameter sets automatically. The tolerance parameters to the left and the right are varied in a range from 0 to 3 seconds using an increment 0.5. The silence penalty is varied in a range from 0 to 2 analogously. The training data is weighted zero or once on utterance level. The action-level weighting is varied between 0 and 5. On the action-object level, weighting factors from 1 to 10 have been explored. We have chosen 12 sets of these factors in order to evaluate models with different degrees of specialisation. All combinations of these parameters result in 2 892 different sets. Each one is used to generate a complete set of action-specific bi-gram language models. Unseen events are handled using absolute discounting with $\beta = 0.8$. Due to the large number of parameter sets and the resulting complexity, this factor has not been made subject to optimisation. Furthermore, the discounting factor has insignificant influence regarding this method as informal tests have shown.

After the action-specific language models have been created the perplexity is computed so that each combination of language model and the underlying parameter set is associated with one. This way the perplexity can be used as optimisation criterion to find the best language model for each type of action. In the following section we present first results gathered using these models during speech processing.

4 Results

The language models' quality is evaluated by assessing the corresponding speech recognition performance. Our speech recogniser uses a standard time synchronous integrated search strategy to weight hypotheses generated by the acoustic model additionally with the language model. We have implemented a strategy, which enables the speech recogniser to switch language models during speech processing

	W_{ACC} %		W_{CORR} %
Action-Specific	65.98	±1.1	68.77
Base Model	69.39	±1.1	71.96
Difference	−3.41		−3.19
Random Usage	48.61	±1.2	51.36

Table 1: Recognition results (expand strategy) using optimised action-specific language models, trained with utterance parts on action-object level only.

Action	Base perp.	Model perp.	Diff
take-cup	20.84	16.55	4.29
take-tea	34.90	16.97	17.93
take-sugar	24.17	14.04	10.12
take-milk	22.68	19.28	3.40
putdown-tea	28.39	9.83	18.56
putdown-cup	23.01	15.11	7.90
putdown-milk	30.48	12.03	18.45
pourin-tea	41.21	11.95	29.27
pourin-sugar	20.39	12.50	7.89
pourin-milk	36.32	12.54	23.78
pourin-water	34.51	16.10	18.41

Table 2: Comparison of the perplexity regarding the action-specific models against the perplexity using a standard bi-gram trained on the whole utterances. The language models are trained with utterance parts on action-object level only.

using a set of switch points. In our case these switch points are generated from the action annotation. Two strategies have been implemented. The *stick* strategy uses exactly the interval borders and a default model when no annotation is available e.g. between two intervals. The *expand* strategy expands each action interval as far as possible so that an action-specific model is always used. All results are computed using a leave-one-out cross validation as described in section 3.4. The audio data belonging to the excluded test subject for each run is used for evaluating the speech recognizer. Afterwards the word accuracy W_{ACC} and the word correctness W_{CORR} are calculated.

In order to see how the degree of specialisation affects the recognition results it is possible to apply restrictions during optimisation. In the following, we

	W_{ACC} %		W_{CORR} %
Action-Specific	70.56	±1.1	73.20
Base Model	69.39	±1.1	71.96
Difference	**1.17**		**1.24**
Random Usage	69.22	±1.1	71.97

Table 3: Recognition results (expand strategy) using optimised action-specific language models, trained using the utterance level always once. Weighting factors have been made subject to optimisation.

Action	Base perp.	Model perp.	Diff
take-cup	20,43	17,59	2,84
take-tea	26,59	25,15	1,44
take-sugar	23,36	18,98	4,38
take-milk	22,68	21,63	1,05
putdown-tea	26,36	20,57	5,80
putdown-cup	22,51	20,91	1,60
putdown-milk	30,46	21,95	8,51
pourin-tea	27,27	22,51	4,77
pourin-sugar	20,33	15,40	4,93
pourin-milk	31,34	25,46	5,88
pourin-water	29,53	24,62	4,91

Table 4: Comparison of the perplexity regarding the action-specific models against the perplexity using a standard bi-gram trained on the whole utterances. The language models are trained using the utterance level always once.

Action	Tolerance [s] left	right	Silence-penalty
take-cup	2.00	1.00	2.00
take-tea	3.00	3.00	0.00
take-sugar	0.00	3.00	1.00
take-milk	3.00	2.50	0.00
putdown-tea	2.50	0.00	0.50
putdown-cup	3.00	0.50	0.50
putdown-milk	0.50	0.00	1.00
pourin-tea	0.50	2.50	1.00
pourin-sugar	0.50	1.00	1.50
pourin-milk	0.00	2.00	0.00
pourin-water	2.50	1.50	0.00

Table 5: Tolerance parameters found by the optimisation process (cp. table 4). The language models are trained using the utterance level always once.

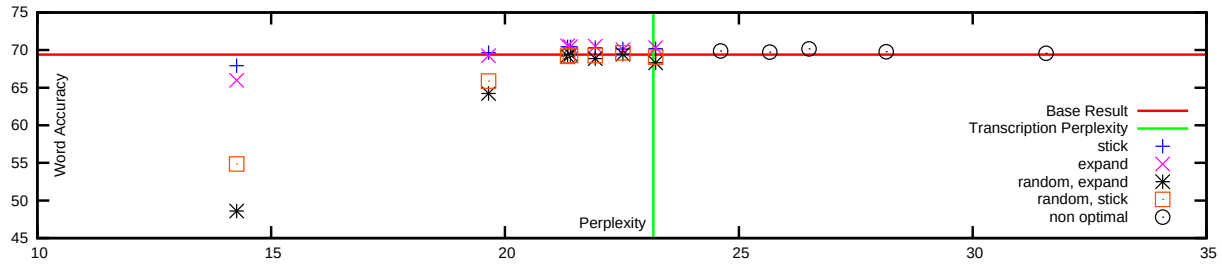

Figure 7: Overview of the average perplexity against word accuracy for all evaluation results. Models that are more specific have a lower perplexity. The difference between random and correct usage is larger for models that are more specific. The optimal results are slightly more specific than the standard bi-gram. Non-optimal models with up to 80 % of the top rated models thrown away do not reach this result. The keywords expand and stick denote the switching strategy where expand means each action interval is expanded as much as possible.

Action	Weighting Factors		
	Utt.	Ac.	Ac.-Obj.
take-cup	1	0	3
take-tea	1	0	1
take-sugar	1	0	3
take-milk	1	0	3
putdown-tea	1	0	5
putdown-cup	1	0	1
putdown-milk	1	1	10
pourin-tea	1	1	5
pourin-sugar	1	1	5
pourin-milk	1	0	5
pourin-water	1	0	3

Table 6: Weighting factors determined during parameter optimisation (cp. table 4).

present detailed results using very specialised models on the one hand and results where the degree of specialisation has also been made subject to optimisation on the other hand. The results are compared against recognition results using a standard bi-gram model trained on the complete utterance level (base result). Another comparison is made against results where an action-specific model is randomly selected for each action interval during speech recognition in order to evaluate their level of specialisation.

Table 1 shows results using very specific models trained with utterance parts on action-object level only. The models are too specific since the results are less good than using a standard bi-gram model. The perplexity difference in table 2 shows that these models are much more specific to the action context than the standard bi-gram model. The random usage result confirms that parts not belonging to the corresponding action context are not well described by the model.

Since very specific models with a low perplexity do not improve recognition results restrictions are applied during optimisation. The results in table 3 are generated using language models, which have been trained using the utterance level always once. The other weighting factors have been made subject to optimisation. The results are significantly better in comparison to the standard model. In contrast to the very specific models, the perplexity difference to the base model is smaller (see table 4). The random usage results emphasise the high level of generalisation. Table 5 shows the optimised tolerance parameters. The according weighting factors are shown in table 6. As one can see, the action-level seems to be of less importance to the specialisation and is therefore rarely used.

We have evaluated more action-specific models optimised under different restrictions. These results are summarized in figure 7. In order to verify that our method actually finds action-specific models which have better results than others trained during the optimisation process we have additionally evaluated non-optimal action-specific models with a lower perplexity. These models are selected by leaving different percentages (from 10 % up to 80 %) of the top rated models unconsidered during the opti-

misation process. The figure shows that these models indeed create worse recognition results than the fully optimised ones.

5 Outlook

We have demonstrated an approach to include visual context into speech recognition realised by means of action-specific language models, which are automatically trained and optimised. The action-specific utterance parts required for training are gained using an automatic associating method between actions and speech. The method only requires manual annotation on a level of low detail. The perplexity is used as optimisation criterion for the training parameter sets and a detailed analysis shows the adequacy of this approach. In order to ensure a certain level of generalisation the complete utterance level has to be always used. The optimisation under this restriction delivers the best results, which are significantly improved in comparison to speech processing with a standard bi-gram model.

Although this approach is able to improve speech recognition, the pairing of speech and actions happens on a heuristic level. Further research has to show in how far this association delivers semantically correct results. In contrast to knowledge-based methods, our approach can easily be transferred to other domains due to the automated pairing and training process.

Further applications of action-specific language models could make it possible that action hypotheses are extracted during speech recognition. In order to realise that, multiple models could be matched against each other during speech processing.

References

James F. Allen. 1983. Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26(11):832–843, November.

G. A. Fink. 1999. Developing HMM-based recognizers with ESMERALDA. In Václav Matoušek, Pavel Mautner, Jana Ocelíková, and Petr Sojka, editors, *Lecture Notes in Artificial Intelligence*, volume 1692, pages 229–234, Berlin Heidelberg. Springer.

A. Green, H. Hüttenrauch, E. A. Topp, and K. S. Eklundh. 2006. Developing a contextualized mulimodal corpus for human-robot interaction. In *Proc. of Int. Conf. on Language Resources and Evaluation (LREC)*, Genua.

Ron Kohavi. 1995. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *International Joint Conference on Artificial Intelligence*, pages 1137–1145.

Zhe Li, Jannik Fritsch, Sven Wachsmuth, and Gerhard Sagerer. 2006. An object-oriented approach using a top-down and bottom-up process for manipulative action recognition. In *DAGM06*, volume 4174 of *Lecture Notes in Computer Science*, pages 212–221, Heidelberg, Germany. Springer-Verlag.

Jan F. Maas and Britta Wrede. 2006. BITT: A corpus for topic tracking evaluation on multimodal human-robot-interaction. In *Proceedings of the international conference on Language and Evaluation (LREC)*, Genoa, Italy.

Harry Mcgurk and John Macdonald. 1976. Hearing lips and seeing voices. *Nature*, 264(5588):746–748, Dezember.

U. Naeve, G. Socher, G. A. Fink, F. Kummert, and G. Sagerer. 1995. Generation of language models using the results of image analysis. In *European Conference on Speech Communication and Technology*, pages 1739–1742, Madrid.

G. Potamianos, C. Neti, G. Gravier, A. Garg, and A. W. Senior. 2003. Recent advances in the automatic recognition of audiovisual speech. *Proceedings of the IEEE*, 91(9):1306–1326.

R. Rosenfeld. 2000. Two decades of statistical language modeling: where do we go from here? *Proceedings of the IEEE*, 88(8):1270–1278, Aug.

Deb Roy and Niloy Mukherjee. 2005. Towards situated speech understanding: visual context priming of language models. *Computer Speech & Language*, 19(2):227–248, April.

M. J. Spivey, M. J. Tyler, K. M. Eberhard, and M.K. Tanenhaus. 2001. Linguistically mediated visual search. *Psychological Science*, 12(4):282–286, July.

S. Wachsmuth and G. Sagerer. 2002. Bayesian Networks for Speech and Image Integration. In *Proc. of 18th National Conf. on Artificial Intelligence (AAAI-2002)*, pages 300–306, Edmonton, Alberta, Canada.

J. C. Wolf and G. Bugmann. 2005. Multimodal corpus collection for the design of user-programmable robots. In *TAROS 2005 Towards Autonomous Robotic Systems Incorporating the Autumn Biro-Net Symposium*, September.

J. C. Wolf and G. Bugmann. 2006. Linking speech and gesture in multimodal instruction systems. In *IEEE International Symposium on Robot and Human Interactive Communication*, pages 141–144, Hatfield, UK, September.