Transfer Learning in Sign language

Ali Farhadi, David Forsyth
University of Illinois at Urbana Champaign
{afarhad2,daf}@cs.uiuc.edu

Ryan White
University of California at Berkeley
ryanw@cs.berkeley.edu

Abstract

We build word models for American Sign Language (ASL) that transfer between different signers and different aspects. This is advantageous because one could use large amounts of labelled avatar data in combination with a smaller amount of labelled human data to spot a large number of words in human data. Transfer learning is possible because we represent blocks of video with novel intermediate discriminative features based on splits of the data. By constructing the same splits in avatar and human data and clustering appropriately, our features are both discriminative and semantically similar: across signers similar features imply similar words. We demonstrate transfer learning in two scenarios: from avatar to a frontally viewed human signer and from an avatar to human signer in a 3/4 view.

1. Introduction

We describe a method for building discriminative word spotters in American Sign Language (ASL). Our method implements a form of transfer learning, where we use one model for phenomena intrinsic to a word and a second model to cover variations in its *rendering* (the signer; the aspect; etc.). Word models can be learned using an animated dictionary, and then spotted in video of a new signer seen from a different aspect. The method is able to transfer models in this way because our features measure similarity between a word and a reference vocabulary.

Sign Language: There is a substantial community of people who are profoundly deaf (an NIDCD report circa 1989 estimates 2 million in the US [2]), of whom perhaps 360,000 speak ASL [1]. It is usual to call this latter group of people Deaf. Very good ASL interpretation services are frequently available in metropolitan areas, but there are many situations in which deaf persons find themselves with inadequate or non-existing interpretation services. For example, there are startlingly few ASL translations of standard diagnostic tests [25, 32].

ASL is rich in complex phonological phenomena [33]. Sign forms can be decomposed into primitive phonological features (evidence includes signs that differ in exactly one feature [22] and "slips of the hand" [18, 22]). Signs can be decomposed into sequences of target positions and movements between these positions; the decomposition obeys constraints [10], which are particularly strong in the case of the non-dominant hand in two-handed signs [6, 12].

Signs are produced more slowly than words (about half the speaking rate), but each sign contains a larger number of features and each feature has a wider range of possible values [22]. Features describing the hands include handshape (for the two hands independently), hand orientation, location of the hand relative to the body, and movement pattern. The Purdue ASL database [26] distinguished 16 different handshapes and 39 movement patterns. Moreover, ASL has an extensive range of "non-manual signals" (NMS), expressed by movements of the torso and head, facial expressions, and eye gaze. These observations have motivated attempts to build multi-channel recognition [38] and multi-channel features [9].

Sign Languages and Computer Vision: Sign languages in general offer important model problems to the vision community. One must recognize phenomena drawn from a very rich, but known, pool; there are important sources of individual variation; there are significant challenges in producing features that are robust to variation in signer, in aspect, and in background.

Authors typically fit Hidden Markov Models to words and use the models discriminatively. Starner and Pentland [31] report a recognition rate of 90% with a vocabulary of 40 signs using a rigid language model. Grobel and Assan recognize isolated signs under similar conditions for a 262-word vocabulary using HMM's [20]. This work was extended to recognize continuous German sign language with a vocabulary of 97 signs by Bauer and Hienz [7]. Vogler and Metaxas use estimates of arm position from a physical sensor mounted on the body or from a system of three cameras and report word recognition accuracy of the order of 90% for a vocabulary of 53 words in [35, 36, 39] and build a phoneme model for 22 word vocabulary without handshapes in [37] and with handshapes in [38]. Kadous trans-

1-4244-1180-7/07/$25.00 ©2007 IEEE

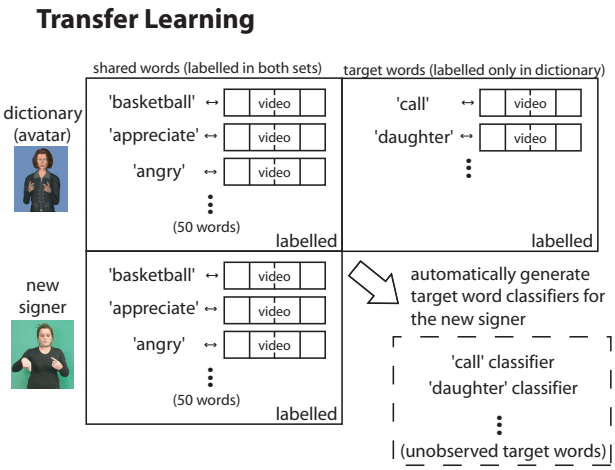

Figure 1. In transfer learning, we use a set of shared words labelled in both datasets and a set of target words labelled only in the dictionary to automatically build word classifiers for the new signer. The resulting word classifiers are highly accurate (Figure 8), even when the avatar and new signer have different appearances (Figure 2).

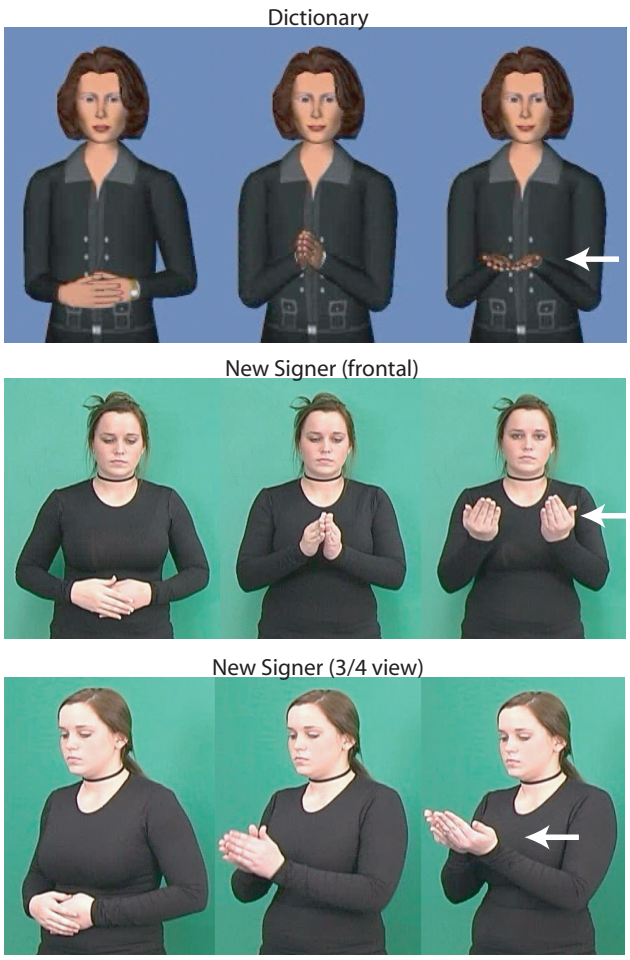

Figure 2. We transfer word models learned on avatar data (**top**) to new domains: a human signer viewed from the front (**middle**) and a human signer recorded in a 3/4 view (**bottom**). Above, we see images from the word 'book' signed in each example. Differing appearance and aspect mean that we can't use the same features. The rightmost frame is indicative of the kind of variations that occur between instances of the same sign. In the frontal view, the human signer holds her hands higher up on the body and further apart than the avatar; in the 3/4 view, she holds her hands higher up than the avatar, but about the same distance apart. Throughout the word, the avatar's gaze is frontal, but the human signer looks at her hands for five of the six frames shown.

duced isolated Australian sign language signs with a powerglove, reporting a recognition rate of 80% using decision trees [28]. Matsuo *et al* transduced Japanese sign language with stereo cameras, using decision tree methods to recognize a vocabulary of 38 signs [27]. Kim *et al.* transduce Korean sign language using datagloves, reporting 94% accuracy in recognition for 131 Korean signs [21]. Al-Jarrah and Halawani report high recognition accuracy for 30 Arabic manual alphabet signs recognized from monocular views of a signer using a fuzzy inference system [4]. Gao *et al.* describe recognizing isolated signs drawn from a vocabulary of 5177 using datagloves and an HMM model [16, 40]. Their system is not speaker-independent: they describe relatively high accuracy for the original signer, and a significant reduction in performance for other signers. Similarly, Zieren and Kraiss report high, but not speaker independent, accuracy for monocular recognition of German sign language drawn from a vocabulary of 152 signs [41]. Akyol and Canzler describe an information terminal which can recognize 16 signs with a high, user-independent, recognition rate; their system uses HMM's to infer signs from monocular views of users wearing coloured gloves [3]. Bowden *et al.* use ICA and a Markov model to learn accurate models of 49 isolated signs using one example per sign [9]. Discriminative word-spotting for a small vocabulary is described in [14]. While a few projects have attempted to translate English into ASL (review in [19]) none have made a heavy use of statistical techniques and only one attempts to align closed captions with ASL [14].

Discriminative methods have not been widely used in studies of ASL (with the exception of [14]). Hidden Markov models are more popular, but in our opinion are not particularly well adapted to modeling sign language. First, unless one has a phonemic dictionary available, one cannot benefit from the pooling of training data across words that is so useful in speech applications — each word model is a completely new model. Second, HMM's are generative and may produce weak results unless one works with features known to be discriminative, particularly when one has few training examples. The advantage of HMM's is their ability to encode dynamical information; as we show, standard discriminative methods can do so perfectly satisfactorily.

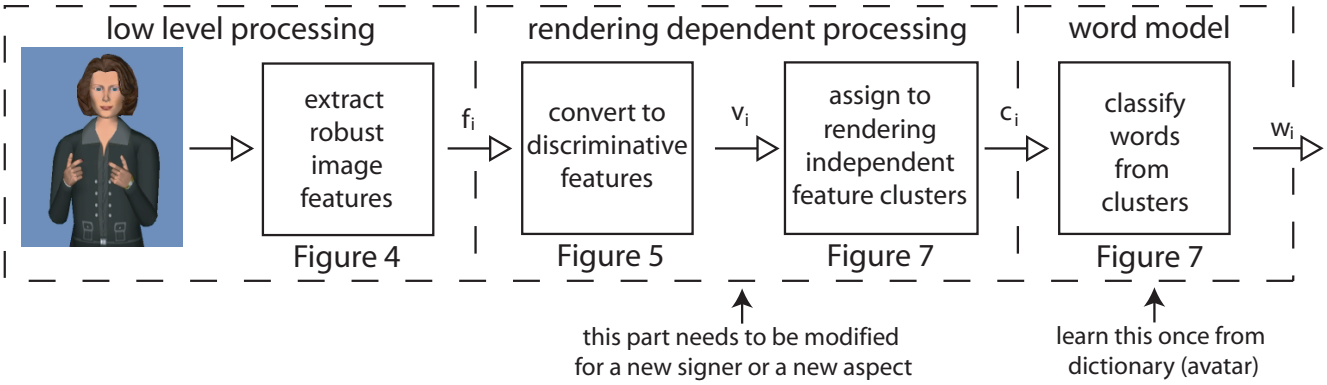

Figure 3. Word spotting is accomplished with three major sections: low level processing, rendering dependent processing, and word modelling. Conceptually, we work backwards across the model. The word model is intrinsic to sign language and only needs to be learned once — therefore the feature clusters c_i in the word model are the same for all videos of sign language. However the image features f_i vary based on the specifics of a particular recording setup. Because these specifics include signer identity and aspect, we can view the middle section as rendering dependent processing. Finally, our image features are based on relatively standard image processing (see Figure 4 for more details).

The great disadvantage of discriminative models to date has been that one may be forced to build word models with relatively few examples (because most words appear seldom, a universal phenomenon in language [24]). Complex aspect phenomena, particularly at the hands, complicate this (e.g. see the collection of aspect information at http://www.bu.edu/asllrp). Worse, for most words, we may have no example of the words produced by the signer in, or at the aspect of, the test sequence. This is a natural application for transfer learning.

Transfer learning describes a body of procedures that allow information obtained learning one task to be transferred to another, related, task. The literature is scattered. One may use empirical priors [17]; "lifelong learning" [34]; determine bias from earlier examples [8]; learn multiple tasks simultaneously [13]; or identify features that tend to transfer [30]. Many vision problems are naturally seen as transfer learning problems (for example, the standard problem of determining whether two face images match without ever having seen images of that individual; as another example, one might use cartoons to learn the location of object features, and very few real images to learn their appearance [11]). This paper demonstrates that, once a semantically similar feature space has been constructed, models of words learned from an animated dictionary alone can be used to recognize words produced by a human signer in different aspects. Figure 1 shows a dataset view of our transfer learning and Figure 2 shows just how different the same word can look.

2. Discriminative Features by Comparison

We wish to recognize signs produced by humans, at novel aspects, from examples produced by a roughly animated signing avatar (which we commonly refer to as the dictionary because of the large number of available words). To do so, we need features that tend to follow intrinsic properties of a sign, rather than the accidents of the rendering of the sign (what the aspect, who the signer, etc.). Our features should not describe what a sign looks like, as this might have to do with the aspect or the signer; it is more useful to describe which other signs, rendered in the same way, look similar to this sign. We expect this form of description to be useful, because phonological studies [37] suggest that signs have shared features — equivalently, that segments of different signs look similar to one another. This fact suggests using comparative features (Section 2.2). In [15], comparative features are used for object recognition. They match

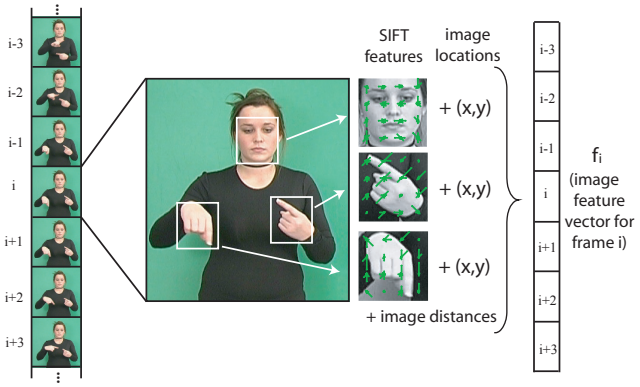

Figure 4. Our image features include motion, image gradient and location information. Each feature is derived from a block of seven frames. In each frame, we cutout both hands and the face to compute sift descriptors. We add position and velocity information (position of the head, offset from one hand to the other and orientation and velocity of each hand) and stack similar data from all seven frames to make one large feature vector.

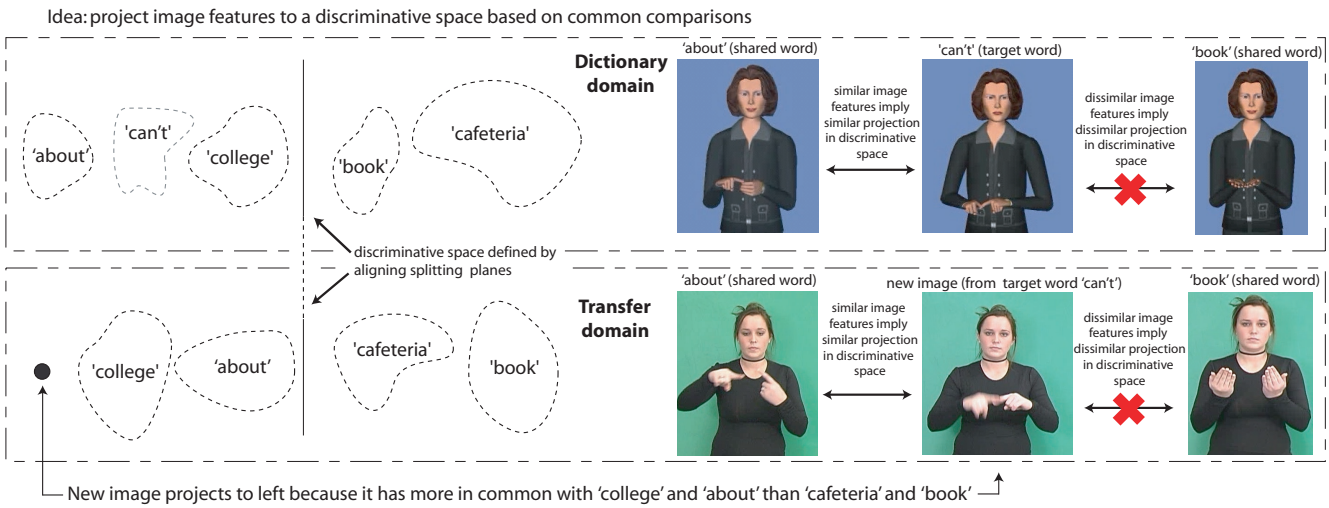

Figure 5. We build discriminative feature spaces from common projections of the image features. Conceptually, the new space allows us to do comparisons in the proper domain. In this figure, we have created two new spaces: one in the dictionary domain (**top**) and one in the transfer domain (**bottom**). Each space is defined by the same split of shared words — in this case the words 'about' and 'college' are separated from the words 'book' and 'cafeteria' (see Section 2 for a description of how we pick splits). Thus, each space has the same semantics and projection operates like a large comparison: image features f_i should project to the side of the decision boundary with similar words. Thus, the word 'can't' has more in common with 'about' than 'book' in both the dictionary and transfer domain. This figure shows a single discriminative projection — in practice we build many such projections and concatenate the results to create a new feature vector. In Figure 3, this corresponds to converting the f_i to the v_i.

images by counting number of common comparisons while we use ordered vectors of comparison results. As a result, their features are far less discriminative because different comparisons reveal different features.

We build our features in two stages. First, we extract an appearance description that represents a block of frames centered at each frame (Section 2.1). Second, we compare these blocks of frames to blocks taken from known example words rendered in the same way (Figure 5); the result of each comparison is binary (Section 2.2).

2.1. Appearance description

Each sign typically spans 25-65 frames of video. Our image features describe a small span of frames (seven in total) centered on a particular frame, to capture local dynamical effects. Figure 4 depicts the extraction of image features. We first identify the hands and head in each frame, using a simple skin detector that uses color thresholds. Skin pixels are clustered to three clusters of known size using k-means. Head and hand configuration is encoded by extracting SIFT features for a bounding box centered on the cluster [23]. When hands overlap, the clusters overlap and so do these bounding boxes, meaning that both hands may be reported with the same feature. This appears to present no problem overall, most likely because hands do not overlap for long periods and we use dynamical features. We always identify the leftmost hand as the left hand. The static feature vector for a single frame is 395 dimensional and consists of SIFT features for head and hands, position of the head, offset from the left to the right hand, and orientation and velocity of each hand. We obtain a dynamic feature vector for each frame by stacking feature vectors for a seven frame interval centered on the current frame. The resulting vector has dimension 2765.

2.2. Comparative features

Our dataset has two types of words: *shared words* that are labelled in both the dictionary and transfer domain and *target words* that are only labelled in the dictionary. We do not expect many shared words or many examples of each; for the work we describe we use 50 shared words and three examples of each.

Each comparative feature is a random split of the shared words into two classes. There are many such splits. A description of the type and utility of these splits is in Figure 5. We choose splits that can accurately be predicted from data and obtain these splits by random search. The features are obtained by using a classifier to predict which side of each split a particular block of frames lies. Recall that our image features represent a block of frames much smaller than a word. As a result, each block of frames of a word must independently project to the same side of the split. This means that good splits tend to be groupings of words with shared structures.

Choosing splits on a dictionary: In detail, we search thousands of distinct randomly selected splits of the shared

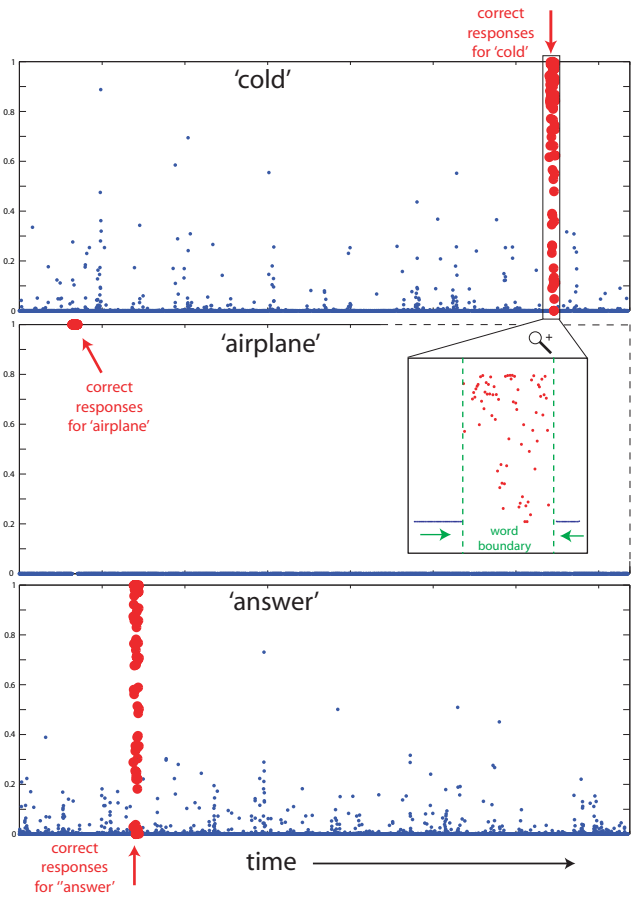

Figure 6. We demonstrate that our discriminative features are useful by building a simple word spotter (trained using logistic regression). In the graph above, responses are in probabilities: 1 indicates the word is present, 0 indicates it isn't. The small blue points are responses to other words and the large red points are responses to the correct word. Notice that the features are discriminative in all cases, and in some cases the response is perfect, as shown in the second graph. See Section 2 for a description of how we build these features. In the inset, we see that the word spotter makes it easy to find word boundaries.

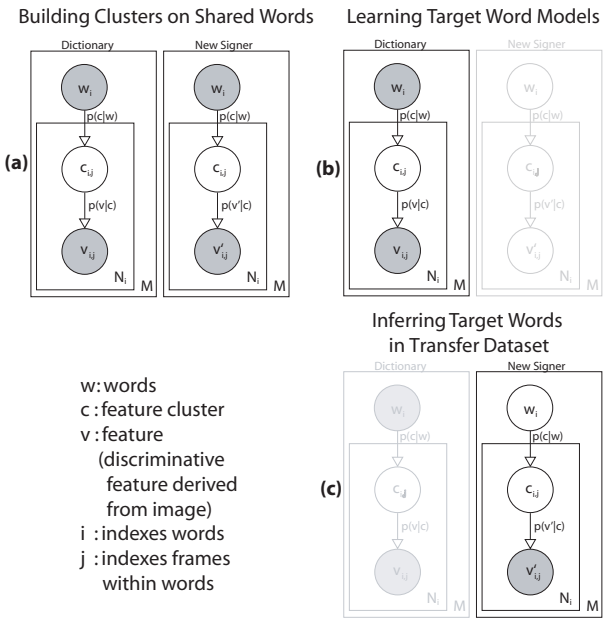

Figure 7. Our word model is a mixture of multinomials. Each word generates a cluster index for each frame of the video. The clusters are groups of discriminative features (derived from image features, see Section 2 and Figure 5). There are actually two models: one for the dictionary and one for the new signer. First, we learn the clusters on shared words using EM. Then, we train target word models on words seen only in the dictionary, and finally we perform inference for these words on video of the new signer.

word vocabulary. For each split, we search random 50-element subsets of the feature vectors *for dictionary renderings alone*. For each such subset, we fit a logistic regression classifier to the split. For each split, we keep the subset of features that produces the logistic regression with the best log-likelihood on the dictionary examples. We now keep the 100 splits with the best log-likelihood on the dictionary.

Learning to split in a new domain: We now have a set of splits that will form the comparative features, but must compute on which side of a split a particular block of frames lies. We have examples of each shared word in each rendering, but the feature vector is too large for reliable classification. For each rendering, we search randomly chosen subsets of 50 elements of the full feature vector, and apply logistic regression to predict the split. We accept the regression that achieves the highest log-likelihood on the training set of frames. We have encountered no overfitting with this model selection strategy, perhaps because the pool of blocks of frames is large.

Knowing the rendering circumstances of any test block of frames (which signer, which aspect ...), we compute the **comparative feature** by evaluating the relevant classifiers for each split on the block and then quantizing the output to zero or one respectively. As a result, each frame of video has a feature vector v_i with 100 binary elements – one element corresponding to each split. These vectors are useful on their own (Section 2.3) or in combination with a word model built for transfer learning (Section 3).

2.3. Spotting Word Boundaries

Our comparative features are highly discriminative *for the rendering for which they were constructed.* We demonstrate this by showing results for a word spotter that uses examples *from the same rendering* to spot new words using the comparative features. Figure 6 shows results of the logistic regression for three words, 'cold', 'airplane', and 'answer', on a long run of 8000 frames. This means that it is straightforward to build a word boundary spotter (using

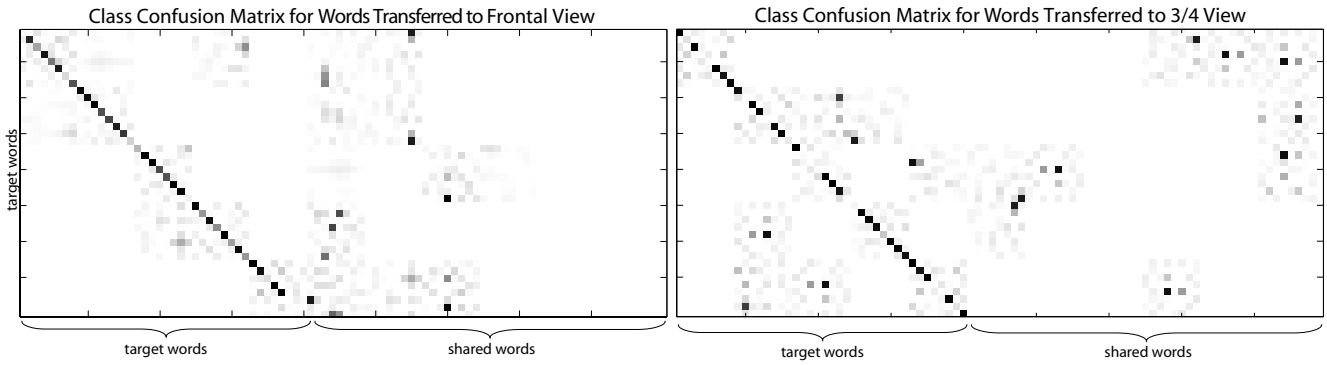

Figure 8. This confusion matrix establishes that transfer learning is accurate, especially considering the difficulty of the task (elements on the diagonal indicate correct classification while others indicate errors). Training for the word model ($p(c|w)$) is done on the avatar and tested on frontal human data (**left**) and 3/4 view human data (**right**). This task is more challenging than a typical application because word spotting is done without language models (such as word frequency priors, bigrams, or trigrams).

logistic regression on comparative features), which is a useful tool. One can then threshold the output of classifier and smooth the results to get the word boundaries (Figure 6). The average error rate for localizing a word boundary using this classifier is 6.78 frames. We use this tool to prepare class confusion matrices (Section 4).

3. Transferable Word Models

To transfer word models from the dictionary to the new space, we use global word models — i.e. word models that remain constant across different signers and different aspects. In other words, there is only one model for the word 'book' in Figure 2 despite significant variation in appearance. We rely heavily on the discriminative features discussed in the previous section.

To accomplish this, we define word models that are a mixture of multinomials (Figure 7) with the following variables: words w_i, discriminative features $v_{i,j}$ as described in the previous section, and a hidden node $c_{i,j}$ which forms clusters of these discriminative features. The model consists of two parts: generating clusters of discriminative features from words ($p(c_{i,j}|w_i)$) and generating specific discriminative feature vectors from clusters ($p(v_{i,j}|c_{i,j})$). The first emission model $p(c_{i,j}|w_i)$ is the word model and is shared for all words. The second emission model $p(v_{i,j}|c_{i,j})$ is specific to the pertinent domain: i.e. we have different $p(v_{i,j}|c_{i,j})$ for different renderings (different signers and different aspects). Correspondingly, in Figure 3, our mixture of multinomials spans two boxes: one for rendering independent processing and one for the word model.

In the following subsections, we discuss a sequence of learning strategies. First, we discuss how to train and use this word model exclusively on the dictionary. Second, we discuss how to perform inference. Finally, we discuss how to train this model so that the dictionary and transfer domain have similar cluster semantics (i.e. c_i are rendering independent).

3.1. Training a Word Model on the Dictionary

In the simplest case, we train our mixture of multinomials on the dictionary alone. We will extend this training in the next section to perform transfer.

In this case, as shown in Figure 7 (b), we have labelled data for the features $v_{i,j}$ and the words w_i, but not the clusters $c_{i,j}$. This is a classic hidden data problem traditionally solved using EM. Because this stage is only an initialization for future steps, we choose a simpler approach: momentarily ignore the word labels, cluster the feature vectors $f_{i,j}$ using k-means and then use the word labels to compute the probabilities ($p(c_{i,j}|w_i)$ and $p(v_{i,j}|c_{i,j})$) by counting.

3.2. Inferring the Word

To infer the word on an unseen sequence of frames, we perform two steps which correspond to the last two steps in our word spotting pipeline (Figure 3). First, we assign the discriminative features ($f_{i,j}$) to clusters probablitically. Second, we sum over this distribution in all frames of the word to obtain an ML estimate of the word identity. As described in Section 2.3, we already know the word boundaries.

3.3. Simultaneous Training in Two Domains

Our word inference procedure is the same in any domain. However, the probabilities $p(v_{i,j}|c_{i,j})$ are specific to the domain and must be trained to force the c_i to be semantically similar between domains. To do this, we initialize both models (dictionary and transfer) using the dictionary model described in Section 3.1. Then we use EM to update the probability models while constraining the word model to be the same in both (i.e. there is one cluster emission model $p(w|c)$ and two feature emission models: $p(v|c)$ for the dictionary and $p(v'|c)$ for the transfer domain). In the

E-step, we compute the expected value for the hidden nodes $c_{i,j}$ and $c'_{i,j}$ and in the M-step we update the probabilities $p(w|c)$, $p(v|c)$, and $p(v'|c)$.

At the end of this procedure we have $p(v'|c)$ for the transfer domain. We can do word spotting by combining the $p(v'|c)$ computed on the shared word and the $p(c|w)$ computed from the target words in the dictionary domain.

4. Results

We demonstrate the effectiveness of our approach by transferring words learned on an avatar to two new domains: a human signer recorded from the front and a 3/4 view.

Data: Figure 2 illustrates our datasets. Our dictionary is an animated ASL dictionary, "SigningAvatar". We exclude synonyms and make a list of 90 words. For each word we render three examples of the word signed at three different speeds. Human signer data consists of three examples of a fluent signer signing the word list in frontal view (the middle row). We obtain one example of each word in a 3/4 view (the bottom row). Word lengths vary from 25 to 65 frames.

Comparative features: We choose 50 shared words to train comparative features. We split these shared words 4000 times and select the 100 best splits. The word model is trained on all 90 dictionary words, and the feature emission model is trained on all 90 dictionary words, but only the 50 shared words for the frontal and 3/4 view signer videos. This means that the remaining 40 words in the human signer videos have not been seen by the word model training process, the feature emission training process, or the comparative feature training process.

Transfer results: Transfer is remarkably successful; results appear to be independent of aspect. In particular, Figure 8 shows class confusion matrices for the 40 words without human signer training data in two cases: a frontal signer and a signer at a 3/4 view. Each target word can be confused with any element of our 90 word vocabulary. In the frontal case, we have 3 instances of each word (and so 120 classification attempts), and we get the error rate of 35.83% which means that 64.17% of these attempts are correct. This is very strong performance for 90-class classification without explicit examples. In the 3/4 case, we have one instance of each word (and so 40 classification attempts), and we get the error rate of 37.5% which means that 62.5% of these classification attempts are correct.

Controls: First, we check how difficult the transfer from avatar to human example is. To do so, we train a word spotter on frontal avatar data and apply it to human examples in frontal and 3/4 view. This gives us the error rate of 99.1% (c.f. our transfer learning error rate of 35.8%) and 97.8% (c.f. our transfer learning error rate of 37.5%) for transferring to frontal and 3/4 view respectively.

Second, we compare dimension reduction with random projection to that using principal components. If we use PCA to compute dimension reduced features rather than random splits, we get an error rate of 64.2% for transferring from frontal avatar to frontal human signer and 68.7% for transferring from frontal avatar to 3/4 view human signer. We conjecture that the increased error rate results from the variance in estimating the PCA projection matrix, which has approximately 3×10^5 entries.

5. Discussion

We have demonstrated that word models learned on an animated avatar can be transferred to spot words produced by a human signer, at frontal and 3/4 views. In particular, the words we spot have never been seen rendered in the form in which we spot them; this is transfer learning.

One application for transfer learning is machine translation. In this setting, a word model such as the one described by this paper is augmented by a language model that includes prior word frequency and local structure (such as bigrams or trigrams). In speech recognition, these additional constraints greatly improve accuracy and we believe that the same is true for ASL.

Our discriminative features based on comparisons are the core of our method. We believe that these features are effective because *comparisons are transferable*. In fact, the existence of a phonological dictionary for ASL [37] reveals the shared structure among words. The success of our comparative features is probably due to this shared structure. We note an analogy with the work of Ando and Zhang [5] where using auxiliary tasks improves the performance of the learning algorithm. One could think of each split as an auxiliary task. It would be interesting to see if Ando and Zhang's method yielded better estimates of the coefficients of our splits with very large quantities of unsupervised data.

Ideally, our discriminative features would convert directly between image features and render-independent features. However, in preliminary experiments we found that the comparative features have somewhat different semantics in different domains. We believe that this is because occasionally word pairs look similar in one domain and different in another. We solve this by using the multinomial mixture model to generate clusters with similar semantics.

The primary intention of this work is to demonstrate the merits of transfer learning with comparative features. As a result, we have not experimented with complex backgrounds, although we expect quite good performance for signers wearing long-sleeved clothing. We speculate that many transfer learning opportunities are available in computer vision. We are currently studying the use of these technologies for activity recognition.

Acknowledgments

This work was supported in part by the National Science Foundation under IIS - 0534837 and in part by the Office of Naval Research under N00014-01-1-0890 as part of the MURI program. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect those of the National Science Foundation or the Office of Naval Research.

References

[1] Demographics. Technical report, Gallaudet Research Institute.

[2] National institute on deafness and communication disorders: A report of the task force on the national strategic research plan. *Federal Register*, 57, 1989.

[3] S. Akyol and U. Canzler. An information terminal using vision based sign language recognition. In *ITEA Workshop on Virtual Home Environments, VHE Middleware Consortium*, pages 61–68, 2002.

[4] O. Al-Jarrah and A. Halawani. Recognition of gestures in arabic sign language using neuro-fuzzy systems. *Artificial Intelligence*, 133(1-2):117–138, December 2001.

[5] R. K. Ando and T. Zhang. A framework for learning predictive structures from multiple tasks and unlabeled data. volume 6, pages 1817–1853, 2005.

[6] R. Battison. *Lexical borrowing in American Sign Language*. Linstok Press, Silver Spring, MD, 1978.

[7] B. Bauer and H. Hienz. Relevant features for video-based continuous sign language recognition. In *AFGR*, pages 440–445, 2000.

[8] J. Baxter. A model of inductive bias learning. *J. AIR*, 12:149–198, 2000.

[9] R. Bowden, D. Windridge, T. Kadir, A. Zisserman, and M. Brady. A linguistic feature vector for the visual interpretation of sign language. In *ECCV*, pages Vol I: 390–401, 2004.

[10] D. Brentari. *A prosodic model of sign language phonology*. MIT Press, Cambridge, MA, 1998.

[11] G. Elidan, G. Heitz, and D. Koller. Learning object shape: From drawings to images. In *CVPR '06*, pages 2064–2071, Washington, DC, USA, 2006. IEEE Computer Society.

[12] K. Emmorey. *Language, Congition, and the Brain*. Lawrence Erlbaum, Mahwah, NJ, 2002.

[13] T. Evgeniou and M. Pontil. Regularized multi–task learning. In *KDD '04*, pages 109–117, New York, NY, USA, 2004. ACM Press.

[14] A. Farhadi and D. Forsyth. Aligning asl for statistical translation using a discriminative word model. In *CVPR*, 2006.

[15] F. Fleuret and G. Blanchard. Pattern recognition from one example by chopping. In *NIPS*, 2005.

[16] W. Gao, J. Ma, X. Chen, et al. Handtalker: a multimodal dialog system using sign language and 3d virtual human. In *Proc. Third Int. Conf. Multimodal Interface*, pages 564–571, 2000.

[17] A. Gelman, J. Carlin, H. Stern, and D. Rubin. *Bayesian Data Analysis*. Chapman and Hall, 1995.

[18] A. Hohenberger, D. Happ, and H. Leuninger. Modality-dependent aspects of sign language production. In R. P. Meier, K. Cormier, and D. Quinto-Pozos, editors, *Modality and structure in signed and spoken languages*. Cambridge University Press, Cambridge, England, 2002.

[19] M. Huenerfauth. A survey and critique of american sign language natural language generation and machine translation systems. Technical report, U. Penn., 2003. TR MS-CIS-03-32.

[20] K.Grobel and M.Assan. Isolated sign language recognition using hidden markov models. In *Proc. Int. Conf. System Man and Cybernetics*, pages 162–167, 1997.

[21] J. Kim, W. Jang, and Z. Bien. A dynamic gesture recognition system for the korean sign language (ksl). *Systems, Man and Cybernetics-B*, 26(2):354–359, April 1996.

[22] E. S. Klima and U. Bellugi. *The signs of language*. Harvard University Press, Cambridge, MA, 1979.

[23] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, November 2004.

[24] C. Manning and H. Schütze. *Foundations of Statistical Natural Language Processing*. MIT Press, 1999.

[25] M. Marschark. *Psychological development of deaf children*. Oxford University Press, 1993.

[26] A. M. Martinez, R. B. Wilbur, R. Shay, and A. C. Kak. Purdue rvl-slll asl database for authomatic recognition of amercian sign language. Proc. IEEE Int. Conf. on Multimodal Interfaces, 2002.

[27] H. Matsuo, S. Igi, S. Lu, Y. Nagashima, Y. Takata, and T. Teshima. The recognition algorithm with non-contact for japanese sign language using morphological analysis. In *Proc. Int. Gesture Workshop*, pages 273–284, 1997.

[28] M.W.Kadous. Machine recognition of auslan signs using powergloves: towards large lexicon integration of sign language. In *Proc. Workshop on the Integration of Gesture in Language and Speech*, pages 165–174, 1996.

[29] S. Nayak, S. Sarkar, and B. Loeding. Unsupervised modeling of signs embedded in continuous sentences. In *IEEE Workshop on Vision for Human-Computer Interaction in conjunction with CVPR*, 2005.

[30] G. Obozinski, B. Taskar, and M. I. Jordan. Multi-task feature selection. Technical report, UC Berkeley, 2006.

[31] T. Starner, J. Weaver, and A. Pentland. Real-time american sign language recognition using desk and wearable computer based video. *PAMI*, 20(12):1371–1375, 1998.

[32] A. Steinberg. Issues in providing mental health services to hearing impaired persons. *Hosp. Community Psychiatry*, (42):380–389, 1991.

[33] W. C. Stokoe, D. C. Casterline, and C. G. Croneberg. *A dictionary of American Sign Language*. Gallaudet University Press, Washington, DC, 1965.

[34] S. Thrun. Is learning the n-th thing any easier than learning the first? In *Advances in Neural Information Processing Systems 8*, 1996.

[35] C. Vogler and D. Metaxas. ASL recognition based on a coupling between HMMs and 3D motion analysis. In *ICCV*, pages 363–369, 1998.

[36] C. Vogler and D. Metaxas. Parallel hidden markov models for American sign language recognition. In *ICCV*, pages 116–122, 1999.

[37] C. Vogler and D. Metaxas. Toward scalability in asl recognition: breaking down signs into phonemes. In *Gesture workshop 99*, 1999.

[38] C. Vogler and D. Metaxas. Handshapes and movements: Multiple-channel american sign language recognition. In *Proc. Gesture Workshop*, pages 247–258, 2003.

[39] C. Vogler, H. Sun, and D. Metaxas. A framework for motion recognition with applications to American sign language and gait recognition. In *IEEE Workshop on Human Motion*, 2000.

[40] W.Gao, J. Ma, J. Wu, and C. Wang. Sign language recognition based on hmm/ann/dp. *Int. J. Pattern Recognition and Artificial Intelligence*, 14(5):587–602, 2000.

[41] J. Zieren and K.-F. Kraiss. Non-intrusive sign language recognition for human computer interaction. In *Proc. IFAC/IFIP/IFORS/IEA symposium on analysis, design and evaluation of human machine systems*, 2004.