Aligning ASL for statistical translation using a discriminative word model

Ali Farhadi, David Forsyth
Computer Science Department
University of Illinois at Urbana-Champaign
{afarhad2,daf}@uiuc.edu

Abstract

We describe a method to align ASL video subtitles with a closed-caption transcript. Our alignments are partial, based on spotting words within the video sequence, which consists of joined (rather than isolated) signs with unknown word boundaries. We start with windows known to contain an example of a word, but not limited to it. We estimate the start and end of the word in these examples using a voting method. This provides a small number of training examples (typically three per word). Since there is no shared structure, we use a discriminative rather than a generative word model.

While our word spotters are not perfect, they are sufficient to establish an alignment. We demonstrate that quite small numbers of good word spotters results in an alignment good enough to produce simple English-ASL translations, both by phrase matching and using word substitution. **Keywords:** *Applications of Vision; Image and video retrieval; Object recognition; Action Analysis and Recognition.*

1. Introduction

Recognition: Recognizing American sign language (ASL) has been well studied. Authors typically fit Hidden Markov Models to words, and use the models discriminatively. Starner and Pentland [17] 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 [8]. This work was extended to recognize continuous German sign language with a vocabulary of 97 signs by Bauer and Hienz [2]. 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 [19, 20, 23], and build a phoneme model for 22 word vocabulary without handshapes in [21] and with handshapes in [22]. Bowden *et al.* use ICA and a Markov model to learn accurate models of 49 isolated signs using one example per sign [3]. There is no explicitly discriminative model in the literature.

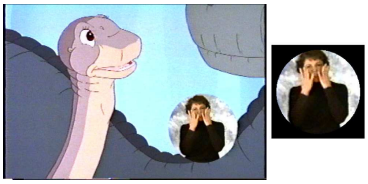

Figure 1. **Left:** *a typical frame, showing the subtitle in a circle on the bottom left corner, one of four possible locations. The extracted ASL frame is on the* **right***, expanded to emphasize the relatively low resolution available.*

Alignment While a few projects have attempted to translate English into ASL (see review in [7]) none have made a heavy use of statistical techniques and the literature contains no attempt to align closed captions with ASL (or any other sign language). Alignment is an established and important tool in the statistical machine translation literature (reviews in [13, 11]). A **bitext** is a body of material in two languages that has the same meaning; the most used example is Canadian Hansard, which reports the doings of the Canadian parliament in both English and French. In alignment, one attempts to find correspondence between increasingly fine units of meaning for a bitext: typically, one aligns first at (roughly) the paragraph level, establishing which paragraphs correspond to which; then at the sentence level, and then, using more sophisticated models of correspondence, at the phrase or word level (it is possible to align quite complex syntactical structures [15]). Coarse alignments can be done linearly [13]. The most stable landmarks for English-ASL alignment are "frozen" ASL signs, which are produced with consistent gestures. These include nouns, certain non-manual features (e.g. the facial expression for questions), and possibly those verbs which are not spatially directed and don't incorporate classifiers (e.g. KNOW and WORK).

A **statistical translation** is obtained from an alignment by choosing a best target sentence, given a source sentence, under various models of word correspondence (methods

Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)
0-7695-2597-0/06 $20.00 © 2006 IEEE

Figure 2. *Continuous ASL video is rich in tricky phenomena.* **Top** *two rows show examples of the word "grandma"; we show every 2nd frame of the word.* **Bottom** *two rows show examples of the word "grandpa"; again, we show every 2nd frame of the word. Note the word length is not constant, and that the role of right and left hands have switched for the two examples of grandpa.*

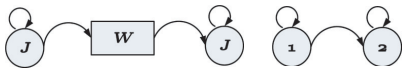

Figure 3. *Rough linear alignment offers pools of continuous ASL video within which a word is known to lie. To build a word model, we need to identify the start and end of the word. Typically, we have three example pools for each word. We fit four HMM's to these examples, two with the state model on the* **left** *(where the word model itself is a 48 state HMM with self-loops and skipping; the two HMM's have different emission models), and two with the HMM on the* **right***, which looks for substantial changes in the sequence. We expect these to occur at the start and the end of the word, and so look for start running this HMM forward in time and end running it backward. This procedure yields three putative start points and three putative end points for each training example; we take the median. Figure 7 shows the method is successful.*

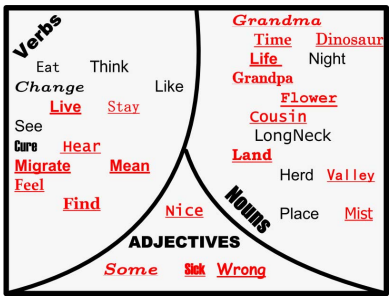

Figure 4. *We have built discriminative word models for a vocabulary of 31 words, shown here broken out by part of speech. The only language model is the transcript which must be aligned. Words in red are words for base words; all other words are derived words, using the language of section 2.*

date to at least [4]; see the review in [1]). Some models come with huge difficulties in identifying the best sentence. An increasingly popular alternative is to work with somewhat larger chunks than words: One cuts-and-pastes from a corpus of aligned bitext, with some limited linguistic transformations (e.g. substituting one word for another word of the same class). This "Example Based Machine Translation" approach dates to at least [14] and has been used successfully by the Diplomat project to quickly develop translators for new languages [5, 6]. While phrasal translation appears to be a natural method for English-ASL translation, we are aware of no attempt to build phrasal translation methods, perhaps because no-one has aligned a bitext. Bungeroth and Ney [24]suggest a statistical machine translation system for transcribed sign language, using a transcription notation like HamNoSys, and written language, especially for the language pair German Sign Language (DGS) and German.

Wordspotting is a method introduced by Rath and Manmatha [16] for identifying words in handwriting; one builds a discriminative word model using aligned training data.

2. Word Spotting with a Discriminative Model

Typically, most words appear seldom [12], and this means that, to align a large body of continuous sign language with a transcript, one must be willing to build word models with relatively few examples. Furthermore, these examples will not be well localized; instead, one is presented with chunks of video within which a word is known to appear. HMM's 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. We are not aware of a phonemic dictionary for any sign language, though there is work on ASL phonology [18] and extensive linguistic work [9]. 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.

2.1. Dataset

Our dataset consists of 80000 frames from an ASL subtitled version of the film "The Land before Time III: Journey through the Mists", digitized from video tape, and with closed captions extracted. There are variations in subtitling practice, some using ASL word order and constructions and others using English word order (which our subtitles do).

Figure 5. *Our film, "The Land before Time III: Journey through the Mists" consists of a series of blocks of video where spoken narration, ASL subtitling and closed captions occur, interspersed with video where there is neither dialogue nor narration. In a typical block, there are several speakers.* **Top:** *A timeline for one such block, with discriminative word model responses marked. The green blocks represent correct detections, the red blocks represent false positives. We are not aware of false negatives. Each green block is linked to the first letter of the word it represents in the transcript* **below** *(where spotted words are marked in green). Figure 6 shows a second such sequence.*

Frames are extracted using a commercial application, FinalCut Express, which can suppress the most objectionable interlacing effects. The subtitling appears in a circle placed in one of four places in each frame, and straightforward color and texture tests are sufficient to tell reasonably reliably whether and where a subtitle appears. Frames without subtitles are discarded, the subtitle is cut out, and the outside of the circle is blacked out (figure 1).

2.2. Rough Alignment

Our film, "The Land before Time III: Journey through the Mists" consists of a series of blocks of video where spoken narration, ASL subtitling and closed captions occur, interspersed with video where there is neither dialogue nor narration. In a typical block, there are several speakers. All this means that video blocks and transcript blocks are already aligned. Within a block, a linear alignment assumes that each word is of fixed length, and for a given word obtains a window of video of typically much greater length centered by word counting. These sequences are expected to contain the word, amid much video "junk". We start with three example sequences per word, each of which consists of 300 frames; our words typically run to 30 frames.

2.3. Features

Subtitle frames are produced as in section 2.1. We then identify possible 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; we use local code, implementing the features of Lowe [10]. 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 classify 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 left to right hand, and orientation and velocity of each hand. We obtain a **dynamic feature vector** for each frame by stacking feature vectors for an 11 frame interval centered on the current frame. The resulting vector has dimension 4345. This dimension is disturbingly large; in section 2.5, we demonstrate a novel method of dimension reduction using discriminative features.

2.4. Finding Word Boundaries for Training

In our experience, HMM's produce relatively poorly discriminative word models under these conditions, but are good at identifying start and end of word within a sequence. However, at this stage we do not have a clear model of which features are discriminative or are best used to produce a word model. For different words, different features seem to apply. We therefore use several HMM's to estimate word start and end, and take the median as an estimate. We use two topologies: a junk-word-junk model (where the word has 48 hidden states, with self-loops and skips) and a 1-2 model (figure 3), which is intended to find significant changes in the sequence. We use the junk-word-junk model with two sets of features. First, we use the per frame static features, dimension reduced using PCA to 30 dimen-

sions. Second, we use the dynamical feature vector, dimension reduced using PCA applied to head and hand shape information to 189 dimensions. Inference by dynamic programming then produces two estimates of the start and end of the word.

Our 1-2 model uses a 44 dimensional feature vector consisting of the velocity measurements for each hand for an 11 frame window centered on the current frame. This model looks for large changes in dynamical structure; we apply it forward and backward in time, to obtain a third estimate of start and end point respectively. We then take the median estimate for start and end point for the word.

2.5. Building Discriminative Word Models

We now have three fairly accurate example windows for each word. We select a subset of B words to serve as **base words**, to which we fit a one-vs-all discriminative model using each frame of each word as examples, and using the dynamic feature. The model is fit using logistic regression on the dynamic features. One could reasonably fear variance problems, given the feature vector is 4345 dimensional and there are relatively few examples; we have not encountered any. However, we do not wish to fit a discriminative model for each word to a feature vector of such high dimension. We therefore use the posterior as a feature.

For any base word, we can compute the posterior that a frame comes from that base word or some other, conditioned on an 11 frame window centered on the frame. We now compute a new set of features for each frame, consisting of the posterior for each base word. We use this B dimensional feature vector as a feature vector for all other words. For other words, we apply logistic regression to the output of the logistic regression that represents base words. The feature vector is attractive, because it is much **lower dimensional** than our original dynamical feature vector; because it is **composed of discriminative features**, albeit discriminative for related tasks; and because it **compares** frames with other words. We call words for which the discriminative model is built in this way **derived words**. Currently, base words are chosen at random, but we believe that some choices are better than others, and are investigating a search procedure.

We can now compute a posterior that any frame comes from a given word or some other, conditioned on a window of frames centered on the current frame. We compute this posterior for each frame and each word that appears in the corresponding transcript block. Detection consists of two stages: First, for each frame declare a detection of a word when its posterior exceeds some threshold. Second, suppress detections that span fewer than half the average size of a training word. We do not require that words exclude one another (meaning that a frame might be allocated to more than one words). The transcript will be used to determine what, if any, word is present.

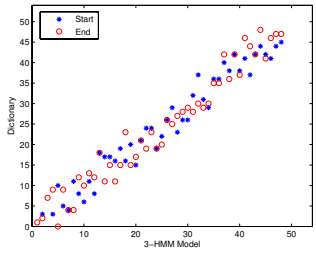

Figure 7. *The scatter plot compares manually determined start and end points found using an ASL dictionary with those obtained in pools of training video by our median method (section 2.4; figure 3) for 48 examples of 16 words (3 examples per word). Our method localizes starts and ends in continuous ASL video to within approximately two frames.*

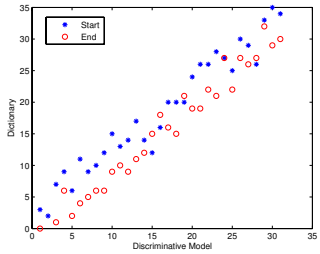

Figure 8. *The scatter plot compares manually determined start and end points found using an ASL dictionary with those obtained by our discriminative method (section 2) for 30 words. Note the offset in the scatter plot, indicating a small bias in the predicted start/end points; we do not currently correct this bias, but believe that it indicates that word boundaries in continuous ASL video may be accompanied by characteristic features that could be found without word models. Allowing for bias correction, our method localizes starts and ends in continuous ASL video to within approximately two frames.*

2.6. Alignment, Word Order and Correction

We now have a block of video with possible word detects in place. The word detectors have not produced false negatives in our experiments, but do produce the occasional false positive (figures 5 and 6). However, the purpose of the exercise is alignment, meaning that a transcript is available. The transcript does not exactly correspond to the ASL (for example, synonyms in the transcript might produce the same signs). However, the nouns, verbs and adjectives in the transcript are generally reproduced in the video and appear in the same order. We use a simple finite-state method to obtain a set of detections of words that are (a) detectable in the transcript and (b) appear in the video in the same order in which they appear in the transcript. A more complex finite-state machine might be required to manage ASL word order; sufficient information about ASL syntax exists

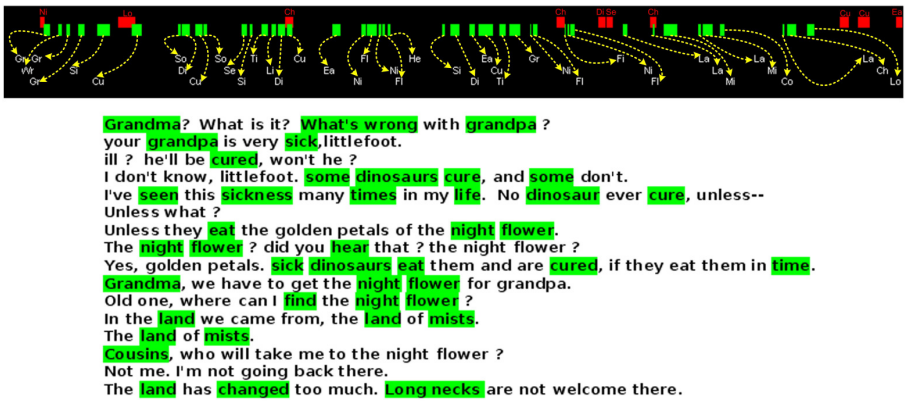

Figure 6. **Top:** *A timeline for a second block, with discriminative word model responses marked. The green blocks represent correct detections, the red blocks represent false positives. We are not aware of false negatives. Each green block is linked to the first letter of the word it represents in the transcript* **below** *(where spotted words are marked in green). Figure 5 shows another such sequence.*

that building such a machine appears practical (e.g. [18]).

3. Results

Precision and recall are not meaningful measures for an alignment; instead, we demonstrate the accuracy of our method with figures giving the partial alignments obtained for two blocks and with example phrasal translations that result.

3.1. Word Spotting

We have checked our method for identifying word boundaries from initial linear alignments using correct boundaries established by hand using the American Sign Language Browser at Michigan State University. Figure 7 shows that our method is extremely effective at pulling out word boundaries from long windows of continuous ASL. We use 21 base and 10 derived words. The discriminative model is accurate at obtaining word boundaries, in comparison to manual estimates (figure 8). made using a good dictionary, though there is some small bias (figure 8). The transcript easily deals with the small number of false positives that occur (figure 5 and 6). Note that the relatively small vocabulary is still large enough to identify phrase boundaries for several useful phrases, and to allow phrasal substitution.

3.2. Phrasal Translation

We are able to isolate several phrases; this means that (a) those phrases can be translated from English to ASL and (b) that these phrases can be employed productively (figure 9). The supplementary material contains ASL translations of the phrases "We have to get the dinosaur"; "What is it"; "Your grandpa is very nice"; "Your Grandpa is very wrong"; "Your nice dinosaur is very sick"; "Your sick dinosaur is very cured"; and "Your sick dinosaur is very nice".

4. Discussion

We have shown that alignment (and so phrasal translation) of a continuous ASL-English bitext is possible using a simple discriminative word model. Our model is novel in not explicitly encoding the dynamical structure of a word. Our use of posterior information for base words as a feature for derived words is novel, and suggests a general method of building image features. Word spotting results are good, and the small errors that do appear can be dealt with by the alignment. Future work will involve building a more sophisticated finite state representation of ASL word order, to allow alignment of more complex ASL sequences.

References

[1] Y. Al-Onaizan, J. Curin, M. Jahr, K. Knight, J. Lafferty, D. Melamed, F.-J. Och, D. Purdy, N. Smith, and D. Yarowsky. Statistical machine translation, final report. Technical report, JHU, 1999. 2

[2] B. Bauer and H. Hienz. Relevant features for video-based continuous sign language recognition. In *AFGR*, pages 440–445, 2000. 1

[3] 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. 1

[4] P. F. Brown, J. Cocke, S. A. D. Pietra, V. J. D. Pietra, F. Jelinek, J. D. Lafferty, R. L. Mercer, and P. S. Roossin. A statistical approach to machine translation. *Computational Linguistics*, 16(2):79–85, 1990. 2

[5] R. Frederking. Interactive speech translation in the diplomat project. *Machine Translation*, 15:27–42, 2000. 2

[6] R. E. Frederking. Speech translation on a tight budget without enough data. In *Proc. Workshop on Speech-to-Speech Translation, ACL-02*, pages 77–84, 2002. 2

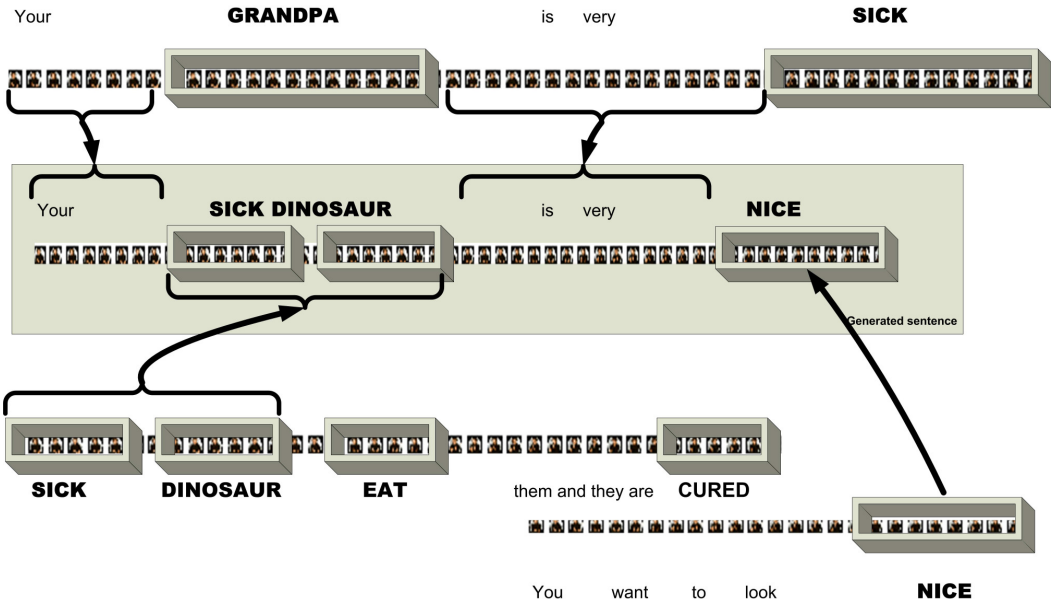

Figure 9. *Phrasal translation of ASL is relatively straightforward with a wordspotter. The relatively small size of our film and its limited vocabulary (together with our small spotting vocabulary) mean that the set of translatable sentences is small compared with English, but is in fact surprisingly large. Here we illustrate taking phrases whose boundaries are spotted in figure 6 (***above** *and* **below** *the shaded block), and then replacing terms (to produce the shaded block in the* **center***). Notice that this phrase combined with our spotting vocabulary, is quite productive: we can produce "your dinosaur is very nice"; "your valley is very nice"; "your nice dinosaur is very sick", "your sick grandpa is very wrong", etc.*

[7] 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. 1

[8] K.Grobel and M.Assan. Isolated sign language recognition using hidden markov models. In *ICSMC*, pages 162–167, 1997. 1

[9] S. K. Liddell. *Grammar, Gesture, and Meaning in American Sign Language*. Cambridge Univ Press, Cambridge, 2003. 2

[10] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, November 2004. 3

[11] E. Macklovitch and M. Hannan. Line'em up: advances in alignment technology and their impact on translation support tools. *Machine Translation*, 13(41-57), 1998. 1

[12] C. Manning and H. Schütze. *Foundations of Statistical Natural Language Processing*. MIT Press, 1999. 2

[13] I. D. Melamed. *Empirical Methods for Exploiting Parallel Texts*. MIT Press, 2001. 1

[14] A. Meyers, M. Kosaka, and R. Grishman. Chart-based transfer rule application in machine translation. In *Proc. 17th conf. Computational linguistics*, pages 537–543, 2000. 2

[15] A. Meyers, R. Yangarber, and R. Grishman. Alignment of shared forests for bilingual corpora. In *Proc. 16th conf. Computational linguistics*, pages 460–465, 1996. 1

[16] T. Rath and R. Manmatha. Word image matching using dynamic time warping. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2003. 2

[17] 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. 1

[18] W. C. Stokoe, D. C. Casterline, and C. G. Croneberg. *A dictionary of American Sign Language*. Gallaudet University Press, Washington, DC, 1965. 2, 5

[19] C. Vogler and D. Metaxas. ASL recognition based on a coupling between HMMs and 3D motion analysis. pages 363–369, 1998. 1

[20] C. Vogler and D. Metaxas. Parallel hidden markov models for American sign language recognition. pages 116–122, 1999. 1

[21] C. Vogler and D. Metaxas. Toward scalability in asl recognition: breaking down signs into phonemes. In *Gesture workshop 99*, 1999. 1

[22] C. Vogler and D. Metaxas. Handshapes and movements: Multiple-channel american sign language recognition. In *GW03*, pages 247–258, 2003. 1

[23] C. Vogler, H. Sun, and D. Metaxas. A framework for motion recognition with applications to American sign language and gait recognition. pages xx–yy, 2000. 1

[24] J. Bungeroth, H. Ney. Statistical Sign Language Translation. Proc. LREC, pages :105–108, 2004. 2