Learning sign language by watching TV (using weakly aligned subtitles)

Patrick Buehler
Engineering Science
University of Oxford, UK
patrick@robots.ox.ac.uk

Mark Everingham
School of Computing
University of Leeds, UK
me@comp.leeds.ac.uk

Andrew Zisserman
Engineering Science
University of Oxford, UK
az@robots.ox.ac.uk

Abstract

The goal of this work is to automatically learn a large number of British Sign Language (BSL) signs from TV broadcasts. We achieve this by using the supervisory information available from subtitles broadcast simultaneously with the signing.

This supervision is both weak and noisy: it is weak due to the correspondence problem since temporal distance between sign and subtitle is unknown and signing does not follow the text order; it is noisy because subtitles can be signed in different ways, and because the occurrence of a subtitle word does not imply the presence of the corresponding sign.

The contributions are: (i) we propose a distance function to match signing sequences which includes the trajectory of both hands, the hand shape and orientation, and properly models the case of hands touching; (ii) we show that by optimizing a scoring function based on multiple instance learning, we are able to extract the sign of interest from hours of signing footage, despite the very weak and noisy supervision.

The method is automatic given the English target word of the sign to be learnt. Results are presented for 210 words including nouns, verbs and adjectives.

1. Introduction

Television programs are now routinely broadcast with both subtitles and a person signing (usually as an overlay) to provide simultaneous 'translation' of the spoken words for deaf people. Our objective is to learn the translation of English *words* to British Sign Language *signs* from this material. We cast the problem as one of (multiple instance) learning from weakly aligned text – an approach that has been increasingly explored for learning visual recognition of objects [8, 14], people [1, 9] and actions [12, 17].

Our aim in this work is, given an English word, to automatically and accurately obtain the corresponding BSL video sequence. We use the English word to select a set of subtitles which contain the word – these form the positive training set – and a much larger set of subtitles that do not contain the word – these form the negative set, see Figure 1.

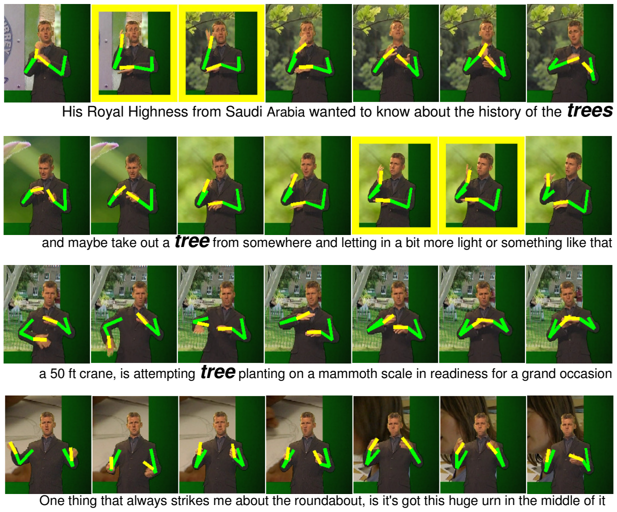

His Royal Highness from Saudi Arabia wanted to know about the history of the ***trees***
and maybe take out a ***tree*** from somewhere and letting in a bit more light or something like that
a 50 ft crane, is attempting ***tree*** planting on a mammoth scale in readiness for a grand occasion
One thing that always strikes me about the roundabout, is it's got this huge urn in the middle of it

Figure 1. **Example training data for the target sign 'tree'**. The top three rows are positive subtitle frame sequences, selected because they contain the text word 'tree'. However, the sign only appears in the first two (outlined in yellow). The final row is an example negative subtitle sequence which does not contain the text word 'tree' and also does not, in fact, contain the sign for tree. Signs are learnt from such weakly aligned and noisy data.

This is a tremendously challenging learning task given that the signing is continuous and there is certainly not a one to one mapping between signs and subtitle words. For example, BSL has a different ordering to English, and there are ambiguities in translation where the same English word may have different meanings and therefore signs, or the same sign may correspond to multiple English words. This introduces a significant correspondence problem between the subtitle text and overlapping video sequence, so the supervision is weak. The difficulties are akin to those encountered in statistical machine translation of written text [18], but here our data set is far smaller than the huge corpora of written language available. Furthermore, the text word can appear in the subtitle, but may not be signed, so the supervision is noisy.

Previous research in sign language recognition has typically required manual training data to be generated for each

978-1-4244-3991-1/09/$25.00 ©2009 IEEE

sign [7, 15, 20, 21, 22] *e.g.* a signer 'performing' each sign in controlled conditions – a time-consuming and expensive procedure. Many such systems have also considered only constrained situations for example requiring the use of data-gloves [23] or coloured gloves to assist with image processing at training and/or test time. Farhadi and Forsyth [10] have considered the problem of aligning an American Sign Language sign with an English text subtitle, but under much stronger supervisory conditions. To some extent we have traded accurate and labour intensive manual training for a labour-free and potentially plentiful supply of training data, but at the cost of moving from strong supervision (*i.e.* manually signed) to weak and noisy supervision. However, our approach is scalable, and imposes no constraints on the signer *e.g.* wearing gloves or a clean background.

Overview. Our source material consists of many hours of video with simultaneous signing and subtitles recorded from BBC digital television. Section 2.1 describes the data and the supervisory information which can automatically be extracted, consisting of 'positive' and 'negative' video sequences. We track the signer, paying special attention to the fidelity of the hand, and generate a feature vector for each frame which describes the position and appearance of the hands – see Section 2.2. The combination of visual descriptors of hand shape and trajectories with weak supervision from the subtitles forms our training data.

Given this training data, our goal is to determine the temporal *window* in the video corresponding to the target English word. The task is cast as a search problem – intuitively we are looking for a window which appears whenever the target word appears in a subtitle, and does not appear otherwise. As discussed above and in Section 2.1 the weak alignment of the subtitles and noise in the supervision means that our notions of 'appears' and 'whenever' must be defined in a soft and error-tolerant fashion. Section 3 describes the *distance function* defined to measure similarity between windows, and how it is adapted to sign-specific visual features by weighting the dominant/non-dominant hands. Section 4 discusses the *scoring function* used to assess the correlation between appearances of a sign and the target word, based on the probability of predicting incorrect labels for the positive vs. negative sequences, incorporating a model of label noise. We cast this problem as one of Multiple Instance Learning (MIL) [19]. The window finally chosen for a sign is that with highest score over a sliding window search of the positive sequences.

As reported in Section 5.1, with this framework we can learn over 100 signs completely automatically.

2. Automatic generation of training data

This section describes how training data is generated from subtitles and video. By processing subtitles we obtain a set of video sequences labelled with respect to a given target English word as 'positive' (likely to contain the corresponding sign) or 'negative' (unlikely to contain the sign). Articulated upper-body tracking and feature extraction are then applied to extract visual descriptions for the sequences.

To reduce the problems of polysemy and visual variability for any given target word we generate training data from the same signer and from within the same topic (*e.g.* by using a single TV program). Even when working with the same signer, the intra-class variability of a given sign is typically high due to 'co-articulation' where the preceding or following signs affect the way the sign is performed, expression of degree (*e.g.* 'very') or different emotions, and varying locations relative to the body.

2.1. Text processing

Subtitle text is extracted from the recorded digital TV broadcasts by simple OCR methods [9] (UK TV transmits subtitles as bitmaps rather than text). Each subtitle instance consists of a short text, and a start and end frame indicating when the subtitle is displayed. Typically a subtitle is displayed for around 100–150 frames.

Given a target *word* specified by the user, *e.g.* "golf", the subtitles are searched for the word and the video is divided into 'positive' and 'negative' sequences. A simple stemming approach is used to match common inflections (*e.g.* "s", "'s", "ed", "ing") of the target word.

Positive sequences. A positive sequence is extracted for each occurrence of the target word in the subtitles. The alignment between subtitles and signing is generally quite imprecise because of latency of the signer (who is translating from the soundtrack) and differences in word/sign order, so some 'slack' is introduced in the sequence extraction. Given the subtitle in which the target word appears, the frame range of the extracted positive sequence is defined as the start frame of the *previous* subtitle until the end frame of the *next* subtitle. Consequently, positive sequences are, on average, around *400* frames in length. In contrast, a sign is typically around 7–13 frames long. This represents a significant correspondence problem.

The presence of the target *word* is not an infallible indicator that the corresponding *sign* is present – examples include polysemous words or relative pronouns *e.g.* signing "it" instead of "golf" when the latter has been previously signed. As measured quantitatively in Table 1 (first and second columns), in a set of 41 ground truth labelled signs only 67% (10 out of 15 on average) of the positive sequences actually contain the sign for the target word.

Negative sequences. Negative sequences are determined in a corresponding manner to positive sequences, by searching for subtitles where the target word *does not* appear. For any target word an hour of video yields around 80,000 negative frames which are collected into a single negative set.

The absence of the target *word* does not always imply that the corresponding *sign* is not present in the negative sequences. This is because different words might be signed similarly, or a sign might be present in the video but not appear in the subtitles (*e.g*. referred to as "it").

In addition to using the English target word, we also exclude subtitles which contain words that can be represented by signs similar to the target sign. For example the sign for "island" is also used for "building", "place" and "town". This information is readily available: the BSL Dictionary [3] contains a list of such polysemes.

2.2. Visual processing

A description of the signer's actions for each frame in the video is extracted by tracking the hands via an articulated upper-body model. Descriptors for the hand position and shape are collected over successive frames to form a *window* descriptor which forms the unit of classification for learning. The temporal length of the window is between 7 and 13 frames, and is learnt.

Upper-body tracking. We use our upper body pose estimator [4] to track the head, torso, arms and hands of the signer. This method uses a 'generative' approach consisting of a layered pictorial structure model. It requires a few frames (around forty) of manual initialization to specify the size of the parts and learn their colour and shape, and then tracking proceeds automatically for the length of the video (around 90K frames).

This method proves capable of robust tracking for long videos, *e.g*. an hour, despite the complex and continuously-changing background (the signer is overlaid on the TV program). Previous approaches to hand tracking have applied skin colour models [5, 13, 11, 20] or sliding window hand detectors [15]. These methods perform poorly when the hands overlap or are in front of the head, and lose track due to the ambiguities that routinely arise, resulting in poor estimation of hand position or unreliable assignment of hands to 'left' or 'right'. By using a full upper-body model and accounting for self-occlusion our method [4] performs robustly. Figure 1 shows example output of the tracker. In the following, we use the wrist position and rectangular region estimated by the tracker for each hand.

Hand shape description. The 'shape' of the hands is extracted by segmentation, and represented by a HOG descriptor [6, 16], see Figure 2. By 'shape' we mean both the silhouette of the imaged hand and the configuration of the fingers, visible in the internal image features of the hand. To deal with cases where the hands are overlapping or touching, descriptors for each hand and also for the pair of hands are extracted in parallel, and combined as described in Section 3. The size of the hand is up to around 80 pixels square for the left and right hand, and 150 pixels for the case of touching hands.

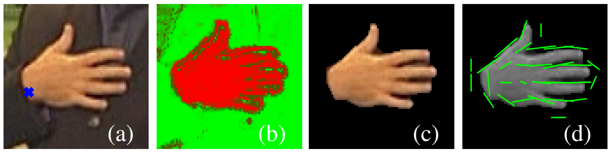


Figure 2. **Segmentation and representation of the hands.** (a) The area around the left hand and estimated position of the wrist (blue cross) obtained from the upper-body tracker. (b) The probability of a pixel being skin (red) or background (green) visualized as RGB components. (c) A graph cut method is used to segment the hand. Note that whole hand is successfully segmented despite pixels around the fingers having a relatively low skin probability. (d) The HOG descriptor for the masked hand image of (c).

A graph cut method [2] is used to segment the hand region predicted by the tracker into hand or background. The unary potential is given by the probability of each pixel being explained by skin or background colour models (Figure 2b). The potential is clamped to background for pixels which are far from the approximate hand rectangle and clamped to foreground for pixels well within the rectangle.

HOG descriptors are chosen for their ability to capture both boundary edges and internal texture, and the contrast normalization they employ gives some invariance to lighting. The HOG descriptor is computed with 4 orientation bins, a cell size of 10×10 pixels and a block size of 1 cell.

Representation by vector quantization. While the raw HOG descriptors could be used to compute similarity between hand shapes, the high dimensionality makes this onerous for large datasets, and errors in the segmentation artificially inflate the distance. In addition, we wish to isolate differences in hand appearance in terms of (i) shape and (ii) orientation. We adopt a vector quantization approach, representing a hand's HOG descriptor by its nearest 'exemplar' hand shape (think visual word). The similarity between hand shapes is then estimated via the distance between the corresponding exemplars, computed off-line and with invariance to rotation (see Section 3).

Representation of hand shape as one of a discrete set is compatible with linguistic analysis of sign language [3], where signs are described as using *e.g*. a "C" or "F" shape taken from the hand configurations used in BSL fingerspelling. Because our descriptor is based on the 2-D image alone, however, we require a considerably larger number of exemplar hand shapes to cover variation in 3-D pose.

Exemplars are learnt separately for the left hand, right hand, and hand pairs, using automatically chosen 'clean' images: the hands must not be in front of the face, and should be separate for individual hands or connected for hand pairs. K-means clustering of the corresponding HOG descriptors is used to determine the exemplar set. We use 1,000 clusters for each of left/right hands and hand pairs.

Given the exemplars, the segmented hands in each frame are then assigned to their nearest exemplar (as measured by Euclidean distance between HOG descriptors) using the position of the wrists in the frame and in the hand exemplar for approximate alignment. Figure 3 shows examples of segmented hands and the exemplars to which they are assigned. Note that by using exemplars extracted from clean images, the correct hand shape is found even in the case where the hands are touching the face.

While facial expression is also used in sign language, we omit its use for the moment since it is challenging to model, and many signs can be distinguished without it.

3. Measuring visual distance between signs

As noted in Section 2.2, our learning approach seeks temporal *windows* of video which represent the same sign, where a window is the concatenation of visual descriptors for a sequence of frames. In order to compare two such windows a distance function is needed which captures differences in position and motion of the hands and their appearance. We assume frame by frame alignment when comparing window descriptors. This is stronger than the assumption typically made in Hidden Markov Models of signs *e.g.* [20], but is justified here because we are generating training data from the same signer.

For each frame t of the window, each hand is described by a vector $\mathbf{x}(t) = \langle \mathbf{x}_{pos}, \mathbf{x}_{hex}, \mathbf{x}_{hex,P} \rangle$ which combines hand position (pos) and the identified hand exemplar (hex) for both the individual hand and the combined hand pair (subscript P). The descriptor for a window $\mathbf{X}$ is the concatenation of the per-frame descriptors $\mathbf{x}(t)$.

In BSL one hand (usually the right) is dominant, while the position and appearance of the other hand is unimportant for some signs. We build this into our distance function. Given two windows $\mathbf{X}$ and $\mathbf{X}'$ the distance between them is defined as the weighted sum of distances for the right (dominant) and left (non-dominant) hands:

$$D(\mathbf{X}, \mathbf{X}') = d_R(\mathbf{X}, \mathbf{X}') + w_L d_L(\mathbf{X}, \mathbf{X}') \qquad (1)$$

where $d_L(\cdot)$ and $d_R(\cdot)$ select the descriptor components for the left and right hands respectively. The weight $w_L \leq 1$ enables down-weighting of the non-dominant hand for signs where it does not convey meaning. Section 4 describes how w_L is learnt for each individual target sign.

The distance measure for the left and right hand alike is defined as a weighted sum over the distances of the position, shape and orientation components (we drop the hand subscript to simplify notation):

$$\begin{aligned} d(\mathbf{X}, \mathbf{X}') \quad = \quad & w_{pos} d_{pos}(\mathbf{X}, \mathbf{X}') + w_{dez} d_{dez}(\mathbf{X}, \mathbf{X}') \\ & + w_{ori} d_{ori}(\mathbf{X}, \mathbf{X}') \qquad (2) \end{aligned}$$

This is in accordance with linguistic sign research [3], where different hand configurations are described separately by shape (d_{dez}) and orientation (d_{ori}).

The positive weights w_{pos}, w_{dez} and w_{ori} are learnt offline from a small number of training examples. In the following we describe each term in the distance measure.

Position distance (d_{pos}). Different repetitions of the same sign vary in position relative to the torso. Especially when a sign is performed in front of the chest, the vertical position in the image can change significantly. In computing the distance between trajectories we therefore introduce some invariance to these effects. The distance is defined as the minimum over a set of feasible transformations:

$$d_{pos}(\mathbf{X}, \mathbf{X}') = \min_T \sum_{t=1}^{n} ||\mathbf{x}_{pos}(t) - T(\mathbf{x}'_{pos}(t))||^2 \qquad (3)$$

where n is the temporal length of the window, $\mathbf{x}_{pos}(t)$ is the (x, y) hand position for frame t and $T(\cdot)$ represents a translation (in both x and y). The maximum translation $T(\cdot)$ in the distance function is again learnt from training data, and is set at 5 pixels. We investigated additional transformations (scaling and rotation), but found this extra invariance to be slightly detrimental to the final performance, for example confusing signs which are mainly distinguished by the orientation of the trajectory.

Hand shape distance (d_{dez}). The distance between hand shapes is designed to account reliably for cases where the hands are either apart or touching each other. When the hands are separated the individual hand shape descriptors are very reliable, however when they overlap, the individual hand shape cannot be reliable determined, and the combined hand pair descriptor should be used. If the hands are not clearly separate, then for a given pair of frames the distance is defined as the minimum over the two cases:

$$d_{dez}(\mathbf{x}, \mathbf{x}') = \min_{\gamma \in \{0,1\}} \gamma d_1(\mathbf{x}, \mathbf{x}') + (1 - \gamma) d_2(\mathbf{x}, \mathbf{x}') \quad (4)$$

where $d_1(\mathbf{x}, \mathbf{x}')$ measures the distance for the hand pair descriptor

$$d_1(\mathbf{x}, \mathbf{x}') = \min_{\phi \in [-90,90]} ||\mathbf{x}_{hex,P} - R_\phi(\mathbf{x}'_{hex,P})||^2 \qquad (5)$$

and $d_2(\mathbf{x}, \mathbf{x}')$ is defined accordingly for the individual hand descriptors by replacing the per-frame descriptors $\mathbf{x}_{hex,P}$ and $\mathbf{x}'_{hex,P}$ in Eqn. 5 by $\mathbf{x}_{hex}$ and $\mathbf{x}'_{hex}$ respectively. The rotation $R(\phi)$ which minimizes the distance between two hand shapes is computed off-line for each pair of exemplars.

The choice of which descriptors to use (γ in Eqn. 4) is made independently for each frame and the window distance is then defined as the sum of distances over all frames:

$$d_{dez}(\mathbf{X}, \mathbf{X}') = \sum_{t=1}^{n} d_{dez}(\mathbf{x}(t), \mathbf{x}'(t)) \qquad (6)$$

In order for the choice between costs in Eqn. 4 to be balanced, the distance on hand pair versus individual hands is normalized appropriately.

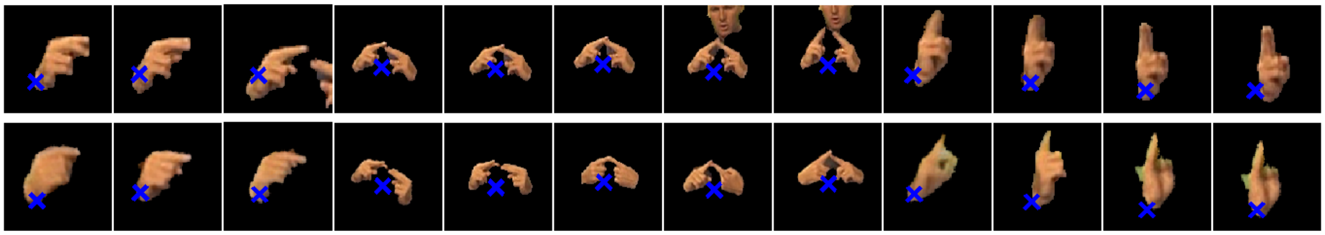


Figure 3. **Examples of hand shape for an instance of the sign "different".** Top row: segmentation results for every frame of the sign (from left to right). Bottom row: identified hand exemplars. If the hands do not touch (computed for each frame) then only the left hand is shown, otherwise the combined hand exemplar is shown.

Hand orientation distance (d_{ori}). The distance between hand orientation for a given frame is defined as the square of the angle needed to rotate one hand exemplar to the other (Eqn. 5). As noted, this is pre-computed for each pair of exemplars. The orientation difference for either the individual hands or hand pair is taken according to the γ value used for the *dez* descriptor for the corresponding frame (Eqn. 4), and summed over all frames of the window.

4. Automatic sign extraction

Given a target word, our aim is to identify the corresponding sign. The key idea is to search the positive sequences to provide an example of the sign. Each positive sequence in turn is used as a 'driving sequence' where each temporal window of length n within the sequence is considered as a template for the sign. The template is scored using the other positive sequences and the negative data. Since the only positive labels available for learning are at the subtitle sequence rather than window level, we can formulate the task naturally as a MIL problem. We require a classifier to determine where the template matches within the sequences, and a score function to assess the contribution from matches within the positive and negative data. The sign is determined by maximizing the score function over all templates and the sign specific dominant/non-dominant hand weighting.

4.1. Sliding window classifier

The classifier is used to determine if a temporal window $\mathbf{X}$ matches the 'template' window $\hat{\mathbf{X}}$. The label for a window $\mathbf{X}$ is predicted by simple thresholding:

$$f(\mathbf{X}, \theta) = \begin{cases} 1 & : \quad D(\hat{\mathbf{X}}, \mathbf{X}) \leq \tau \\ 0 & : \quad otherwise \end{cases} \tag{7}$$

where a positive output indicates that the window represents the sign, and the vector θ specifies all the parameters of the classifier: the threshold τ and the weight w_L.

We next introduce the MIL formulation, and the scoring function.

4.2. Multiple Instance Learning method

For a given target word we are provided with a set of positive *bags* $\mathcal{B}^+ = \{B_1^+, \dots, B_{n_+}^+\}$, where each bag corresponds to a sequence within which the target word appears in the subtitles (Section 2.1). Each bag consists of a set of *instances* $B_i^+ = \{\mathbf{X}_1^i, \dots, \mathbf{X}_{n_i}^i\}$, with each instance being a window descriptor (Section 2.2). Conceptually we assume that a bag should be labelled positive (contains the target sign) if any of the instances represent the sign, and thus we can predict the bag label from the instance labels predicted by the classifier as:

$$f(B_i, \theta) = \begin{cases} 1 & : \quad \exists j : f(\mathbf{X}_j^i, \theta) = 1 \\ 0 & : \quad otherwise \end{cases} \tag{8}$$

where n_i is the number of instances (windows) in the bag (sequence) B_i. As is usual in an MIL formulation, this construction avoids the need for per-instance labels.

In addition we have a set of negative instances $\mathcal{X}^- = \{\mathbf{X}_1^-, \dots, \mathbf{X}_{n_-}^-\}$ which is the union of all windows in the video far from the target word according to the subtitles (Section 2.1). Note that there is an 'asymmetry' here in that the negative data is not naturally aggregated into bags.

For a hypothesized setting of the classifier parameters θ, we assign a 'score' $S(\theta)$ to the classifier as a function of (i) its predictions on the positive bags and negative instances, and (ii) our prior knowledge about the likely temporal location of target signs

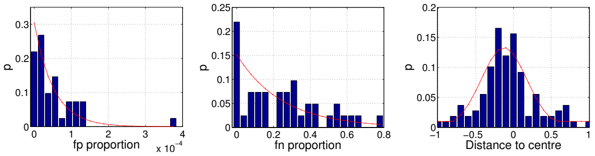
$$S(\theta) = S_+(\mathcal{B}^+, \theta) + S_-(\mathcal{X}^-, \theta) + S_t(\mathcal{B}^+, \theta) \tag{9}$$

and seek the value of θ which maximizes this score.

4.3. Score function for noisy labels

Unfortunately we know that some non-negligible proportion of our 'ground truth' labels obtained via the subtitles will be incorrect, *e.g.* in a positive bag the target word appears but the corresponding sign is not present, or in the negative data the target sign is present but not the corresponding target word. A model of such errors must be incorporated to achieve successful learning.

Since we have no prior knowledge of *which* positive bags or negative instances are likely to have incorrect labels we score the classifier in terms of the *number* of apparent errors made on the training data. The score is defined as the probability of predicting a particular proportion of 'incorrect' labels with the correct classifier, given a model of the



(a) Positive bags (b) Negative bags (c) Temporal prior

Figure 4. **Distributions used to score template windows.** Plots (a) and (b) show the empirical distribution of errors (bars) and the fitted exponential distribution (curve). Note the scale on the x-axis. Plot (c) shows the temporal distribution of signs within corresponding positive sequences.

proportion of errors in the 'ground truth' labels:

$$S_+(\mathcal{B}^+, \theta) \; = \; \log p \left(\frac{1}{n_+} \sum_{i=1}^{n_+} 1 - f(B_i^+, \theta) \middle| \lambda_+ \right) \quad (10)$$

$$S_-(\mathcal{X}^-, \theta) \; = \; \log p \left(\frac{1}{n_-} \sum_{j=1}^{n_-} f(\mathbf{X}_j^-, \theta) \middle| \lambda_- \right) \quad (11)$$

where the sum operation simply counts the number of negative/positive predictions respectively. The distribution $p(r|\lambda)$ models the probability of there being a proportion of r errors in the labels in the positive bags/negative instances.

Modelling the distribution of errors. For positive bags, $p(r|\lambda_+)$ is estimated by fitting a parametric model to ground truth training data. For a subset of target signs we count the number of errors (missing signs) in the subtitle sequences (positive bags). Figure 4a shows a histogram of the proportion of errors for positive bags. The distribution very approximately follows an inverse power law, and we therefore adopt an exponential model:

$$p(r|\lambda) = \lambda e^{-\lambda r} \quad (12)$$

The single parameter λ is fitted by maximum likelihood estimation.

It is difficult to estimate the distribution of errors for the negative data since this requires ground truth labelling of the entire video. We use a heuristic: "for each sign missing from a positive sequence we assume it is seen somewhere in the negative data." Figure 4b shows the resulting distribution – again an exponential model fits the data well. While there will be other causes of errors in the negative data *e.g.* signs which represent more than one word, this heuristic gives good results and we found that our results are not sensitive to the parameter λ_- within an order of magnitude.

With the estimated parameters λ_+ and λ_- the expected proportion of errors in positive labels is 0.25. For the negative data the expected proportion is 5×10^{-5}. Given a negative set of 80000 frames this corresponds to around 4 'false' positives expected if the correct sign is learnt.

It is worth noting that most previous work on modelling errors in labels has assumed a Bernoulli model *i.e.* that labels are independently 'flipped' according to some fixed probability, resulting in a binomial model of proportions. We tried this model but found that it gives a poor fit to the empirical distribution of errors, which has a heavier tail.

4.4. Temporal prior

The sign instances which correspond to a target word are more likely to be temporally located close to the centre of positive sequences than at the beginning or end. We model this by positively scoring classifiers which make predictions close to the centre of the positive sequences:

$$S_t(\mathcal{B}^+, \theta) = \log \prod_{i=1}^{n_+} S_t(B_i^+, \theta) \quad (13)$$

For bags where the classifier output is negative a uniform distribution $S_t(B_i^+|\theta) = 1/n_i$ is used. For other bags the maximum likelihood of the temporal locations of positive predictions is used:

$$S_t(B_i^+, \theta) = \max_{\{j : f(\mathbf{X}_j^i, \theta) = 1\}} p_t \left(\frac{2j}{n_i} - 1 \right) \quad (14)$$

where temporal locations in a sequence are scaled to lie in the range $[-1, +1]$.

The temporal prior p_t is learnt from a subset of signs as for the score functions. As shown in Figure 4c, a Gaussian model gives a good fit to the empirical distribution.

4.5. Searching for the sign by maximizing the score

Given a template window $\hat{\mathbf{X}}$ of length n from a positive sequence, the score function is maximized by searching over the weight for the left hand w_L, and a set of thresholds τ. This operation is repeated for all such template windows, and the template window that maximizes the score is deemed to be the sign corresponding to the target word.

Using a per-sign window length allows for some signs being significantly longer than others. The weight w_L allows the importance of the left hand to be down-weighted for signs which are performed by the right hand alone.

The optimization is carried out as a grid search with the parameter space discretised into 3 different window lengths of $n \in \{7, 10, 13\}$ frames, 4 values for down-weighting the left hand $w_L \in \{1, 0.5, 0.25, 0\}$, and a search over all relevant thresholds, *i.e.* one threshold per positive bag.

Given an average of 15 positive sequences per word, an average of 400 frames per positive sequence, and 3 different window lengths, the number of template windows considered is around 18,000.

5. Experiments and Datasets

Given an English word our goal is to identify the corresponding sign. We deem the output a success if (i) the selected template window, *i.e.* the window with the highest score, shows the true sign (defined as a temporal overlap of at least 50% with ground truth) *and* (ii) at least 50% of all windows within the positive sequences which match the template window show the true sign.

If the window with highest score is not the true sign, we examine the second, third etc. (up to 20) ranked windows by the scoring function to determine which, if any, are correct.

Datasets. We tested our approach on 10.5 hours of signing sequences recorded from BBC broadcasts (including subtitles), aired between 2005 and 2006, and covering such diverse topics as politics, motoring and gardening. Signing is performed by three different persons. The image size after cropping the area around the signer is 300 × 330 pixels.

Test set. The method is evaluated on 210 words. These words were selected and fixed before running the experiments, without knowledge of the appearance of the target signs, *i.e.* how the corresponding sign is performed. Selection was based on: (i) the target word must occur more than 5 times in the subtitles; (ii) the target word is a verb, noun or adjective as opposed to linking words such as "then", "from", "the", etc.; (iii) the target word does not have multiple meanings (as opposed to *e.g.* the word 'bank').

For 41 of these words we carry out an extensive annotation, determining where all the occurrences of the target sign (if any) are in all positive sequences. These signs are listed in Table 1 together with the number of positive subtitle sequences, and the number of times that the target sign actually occurs at least once within these sequences.

The full list of signs used is given at `www.robots.ox.ac.uk/~vgg/research/sign_language/`, which also contains example sequences of the detected signs.

5.1. Results

Results for the 41 word dataset. In the following we report quantitative results for the 41 word dataset for which more extensive ground truth was annotated (see Table 1).

For 30 of these 41 words, we are able to automatically find the corresponding sign. For a further 3 the second best estimate corresponds to the correct sign, and for 1 the third best estimate. The remaining 7 signs are not within the best 3 estimates.

We achieve good results for a variety of signs: (i) signs where the movement of the hand is important *e.g.* "golf", (ii) signs where the hands do not move but the hand shape is important *e.g.* "animal", (iii) signs where both hands are together to form a sign *e.g.* "plant", (iv) signs which are finger spelled *e.g.* "bamboo", and even (v) signs which are performed in front of the face *e.g.* "visitor", which makes identifying the hand shape difficult.

Some of the mistakes are understandable: For the word "wood", our rank 1 result is the sign for "fire". This is not surprising since these two words often appeared together. The sign "year" is difficult since the signs for "last year", "this year" and "next year" differ – our method picks the sign for "next year". In the three cases where the true sign does not lie in the top 20 ranked windows, the proportion of positive sequences which actually contain the target sign is ≤ 0.5.

Sign	n_+	GT	Rank	TP	FP
animal	14	13	1	10	0
auction	7	6	1	4	1
bamboo	20	15	1	14	0
boy	9	3	×		
cabbage	22	21	1	11	0
charcoal	19	16	1	11	1
children	7	7	1	5	1
clothing	7	7	1	5	0
cut	10	7	1	4	0
different	11	10	1	3	1
flower	26	23	1	3	3
fungi	11	9	1	10	0
gang	8	4	×		
garden	24	15	2	9	0
golf	9	9	1	9	0
great	13	9	1	4	1
help	7	5	2	4	0
house	23	14	5	5	0
identification	9	4	1	4	1
island	9	7	11	3	0
kew	40	31	1	29	0

Sign	n_+	GT	Rank	TP	FP
know	9	2	×		
navy	7	7	1	7	0
new	12	5	1	7	1
office	6	5	1	3	0
old	11	8	2	5	1
plant	51	21	1	17	0
prince	10	9	1	6	0
seed	18	15	1	16	0
series	6	5	1	4	0
species	23	14	5	4	1
student	15	12	1	9	0
think	28	21	3	8	2
three	9	4	1	2	2
tree	28	19	1	10	0
visitor	7	4	5	3	1
wood	9	7	5	5	0
work	18	8	1	6	2
world	10	5	1	5	1
wreath	11	9	1	6	0
year	25	14	1	7	0
MEAN	15	10		7	0.5

Table 1. **Ground truth and performance for 41 words.** The first two columns show statistics of the training data; the last three show the performance of the method on these words. n_+ refers to the total number of positive sequences, while GT shows the number of positive sequences in which the target sign occurs. The result has rank 1 if the window with the highest score correctly identifies the sign, rank 2 if the correct window has the second highest score, etc. An × indicates that none of the top 20 ranked windows are correct. The number of correctly detected target signs in the positive sequences is given in TP, while FP gives the number of incorrect detections.

For 30 out of the 41 words, the weight of the left hand w_L is learned to be greater than zero. Out of these, 26 signs are performed with two hands (referred to in BSL as double-dez). In contrast, for the remaining 11 signs where $w_L = 0$, only 6 signs are performed with both hands.

Results for full 210 word dataset. For the full 210 word dataset, in 136 cases (65%) we are able to automatically find the template window which corresponds to the target sign (see Figure 6 for two examples).

The precision-recall curve in Figure 5 (blue dashed line) shows that the score associated with a template window can be used as a confidence measure, giving an extremely good guide to success: at 11% recall (23 signs) precision is 100%; at a recall of 50% (105 signs) the precision is 77%.

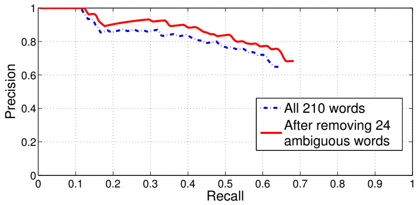


Figure 5. **Precision recall curve** computed using the score of the template window to rank learned signs.

Some words in our dataset co-occur with other words in the subtitles *e.g.* "prince" and "charles", which renders the correct template window ambiguous. Often these incorrectly-learned signs have a high associated score and

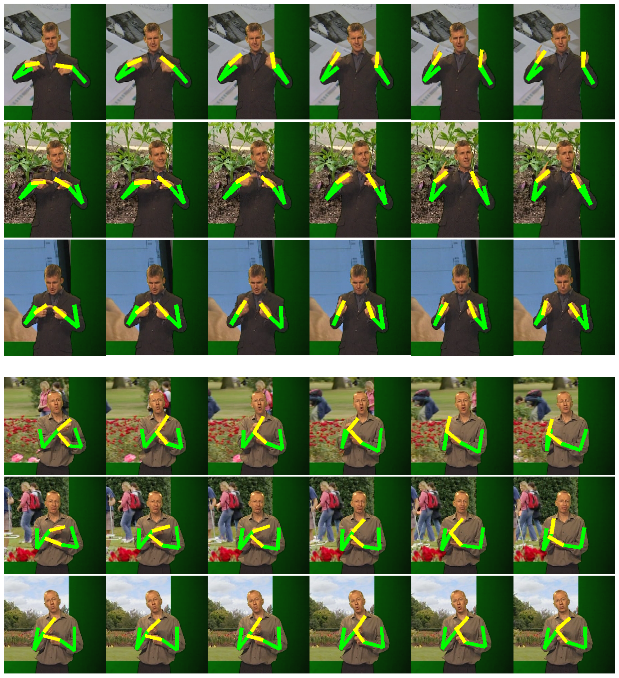


Figure 6. **Example sequences for the signs "different" and "watering"** performed by two different signers and learned automatically. In each figure, the first row shows the template window with the highest score; rows 2 and 3 show identified instances of the sign within the positive sequences.

hence reduce the precision even at low recall. By using simple statistics generated from the subtitles we can identify 24 words for which we expect the proportion of correctly identified signs to be low. Indeed, out of these 24 signs only 36% are learned correctly, as opposed to the average accuracy of 65%. Figure 5 (red solid line) shows the precision-recall curve for the remaining 186 signs.

We evaluated the influence of various components of our method. Without the temporal prior the accuracy decreases from 65% to 57%, and without searching over weights for the non-dominant hand the accuracy decreases from 65% to 59%. We also investigated a voting scheme to select the learnt template window, instead of selecting the window with highest score, and tried learning all the weights $\{w_{pos}, w_{dez}, w_{ori}\}$ in Eqn. 2 for each individual sign. Neither change improved results.

6. Conclusions

We propose a framework based on multiple instance learning which can learn a large number of British Sign Language signs from TV broadcasts. We achieve very promising results even under these weak and noisy conditions by using a state-of-the-art upper-body tracker, descriptors of the hands that properly model the case of touching hands, and a plentiful supply of data. A similar method could be applied to a variety of fields where weak supervision is available, such as learning gestures and actions.

Acknowledgements: We are grateful for financial support from RCUK, EPSRC, the Royal Academy of Engineering, and ONR MURI N00014-07-1-0182.

References

[1] T. Berg, A. Berg, J. Edwards, M. Mair, R. White, Y. Teh, E. Learned-Miller, and D. Forsyth. Names and faces in the news. In *Proc. CVPR*, 2004.

[2] Y. Y. Boykov and M. P. Jolly. Interactive graph cuts for optimal boundary and region segmentation of objects in N-D images. In *Proc. ICCV*, 2001.

[3] D. Brien. *Dictionary of British sign language*. Faber and Faber, 1993.

[4] P. Buehler, M. Everingham, D. P. Huttenlocher, and A. Zisserman. Long term arm and hand tracking for continuous sign language TV broadcasts. In *Proc. BMVC.*, 2008.

[5] H. Cooper and R. Bowden. Large lexicon detection of sign language. *ICCV, Workshop Human Comp. Inter.*, 2007.

[6] N. Dalal and B. Triggs. Histogram of oriented gradients for human detection. In *Proc. CVPR*, 2005.

[7] P. Dreuw, T. Deselaers, D. Rybach, D. Keysers, and H. Ney. Tracking using dynamic programming for appearance-based sign language recognition. In *Face and Gesture*, 2006.

[8] P. Duygulu, K. Barnard, J. F. G. de Freitas, and D. A. Forsyth. Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. In *ECCV*, 2002.

[9] M. Everingham, J. Sivic, and A. Zisserman. "Hello! My name is... Buffy" – automatic naming of characters in TV video. In *Proc. BMVC.*, 2006.

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

[11] A. Farhadi, D. Forsyth, and R. White. Transfer learning in sign language. In *Proc. CVPR*, 2007.

[12] S. Gupta, J. Kim, K. Grauman, and R. J. Mooney. Watch, listen & learn: Co-training on captioned images and videos. In *Proc. ECML PKDD*, 2008.

[13] E. Holden, G. Lee, and R. Owens. Automatic recognition of colloquial Australian sign language. In *Workshop on Motion and Video Computing*, 2005.

[14] J. Jeon, V. Lavrenko, and R. Manmatha. Automatic image annotation and retrieval using cross-media relevance models. In *Proc. ACM SIGIR*, 2003.

[15] T. Kadir, R. Bowden, E. J. Ong, and A. Zisserman. Minimal training, large lexicon, unconstrained sign language recognition. In *Proc. BMVC.*, 2004.

[16] H. Kjellström, J. Romero, D. Martínez, and D. Kragić. Simultaneous visual recognition of manipulation actions and manipulated objects. In *Proc. ECCV*, 2008.

[17] I. Laptev, M. Marszałek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *Proc. CVPR*, 2008.

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

[19] O. Maron and T. Lozano-Pérez. A framework for multiple-instance learning. In *NIPS*, 1998.

[20] T. Starner, J. Weaver, and A. Pentland. Real-time American sign language recognition using desk- and wearable computer-based video. *IEEE PAMI*, 1998.

[21] C. Vogler and D. Metaxas. ASL recognition based on a coupling between HMMs and 3D motion analysis. In *Proc. ICCV*, 1998.

[22] C. Vogler and D. Metaxas. Handshapes and movements: Multiple-channel American sign language recognition. In *Gesture-based Communication in HCI*, 2003.

[23] C. Wang, W. Gao, and J. Ma. A real-time large vocabulary recognition system for Chinese sign language. In *Gesture and Sign Language in HCI*, 2001.