Sign Language Recognition using Linguistically Derived Sub-Units

Helen Cooper, Richard Bowden

University Of Surrey
Guildford, UK.
H.M.Cooper@Surrey.ac.uk, R.Bowden@Surrey.ac.uk

Abstract

This work proposes to learn linguistically-derived sub-unit classifiers for sign language. The responses of these classifiers can be combined by Markov models, producing efficient sign-level recognition. Tracking is used to create vectors of hand positions per frame as inputs for sub-unit classifiers learnt using AdaBoost. Grid-like classifiers are built around specific elements of the tracking vector to model the placement of the hands. Comparative classifiers encode the positional relationship between the hands. Finally, binary-pattern classifiers are applied over the tracking vectors of multiple frames to describe the motion of the hands. Results for the sub-unit classifiers in isolation are presented, reaching averages over 90%. Using a simple Markov model to combine the sub-unit classifiers allows sign level classification giving an average of 63%, over a 164 sign lexicon, with no grammatical constraints.

1. Introduction

Sign Language Recognition (SLR) has many parallels to speech recognition, the idea which has been seized by many is that of combining sub-units into word level classifiers. Doing this has several advantages; it allows the lexicon to be increased in a manageable manner. It removes much of the temporal variance between repetitions of the same sign. It enables linguistics to be used, to add priors to the sub-unit combinations and it could feasibly lead to classification of unseen signs based on their component parts and a dictionary. For these last two advantages to be realised, the sub-unit classifiers need to be derived from the linguistic domain.

Previous systems using tracking-based, sub-unit classifiers, have tended to either hard code basic sub-units (Kadir et al., 2004) or used data driven approaches (Han et al., 2009; Yin et al., 2009). While both these techniques can give good sign level results, they bear little relation to the linguistics of sign language. Instead, the sub-unit classifiers proposed in this paper are learnt from data, annotated at the sub-unit level, using the same notation as that in the British Sign Language (BSL) Dictionary (British Deaf Association, 1992).

In this work, first the signer is tracked, then sub-unit, classifiers are learnt using boosting. Specifically, sub-units relating to Position (*Tab*), Hand Arrangement (*Ha*) and Movement (*Sig*) are covered. These classifier responses are also shown in combination with a Markov chain Look Up Table (LUT) to perform basic classification at the sign level. The details of these classifiers are shown in the following sections.

2. Method

Tracking results are obtained using Buehler *et al.*'s tracker which does not require coloured gloves, whilst still giving accurate results, on natural sign from TV broadcasts it achieves >80% (Buehler et al., 2008). The tracking system gives boxes bounding the hands, lower arms and upper arms. The different sub-unit types are catered for by different weak classifier concepts; *Tab* requires information about positioning, *Ha* about the relationship between the hands and *Sig* about the temporal changes in hand positions, often relative to each other. Each of the different weak classifier types are combined using AdaBoost (Freund and Schapire, 1995) to create a classifier for each sub-unit present in the training set.

2.1. *Tab* Classifiers

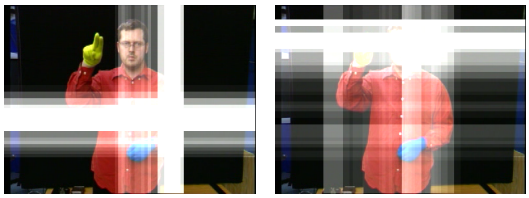
Classifiers concentrating on *Tab* sub-units are concerned with spatial features, describing the location of the hands in relation to the signer. The bounding boxes of the hands are given by the tracking and the position of the face can be found using the Viola Jones face detector (Viola and Jones, 2001). Classifiers can then be built which consider relational distances. Each classifier operates on an x or y feature, i, within the tracking vector, $\mathbf{o}$, comparing it to an upper and lower limit, $\mathcal{T}_U$ and $\mathcal{T}_L$ respectively. If the value falls within this range, then the classifier fires. The upper and lower limits are individual to each classifier and calculated relative to the size (f) and position (f_{xy}) of the signer's face, see Equation 1.

$$\begin{aligned} \mathcal{T}_L &= f_{xy} + nf \\ \mathcal{T}_U &= \mathcal{T}_L + sf \\ n &\in \{-3, -2.9, -2.8 \dots 3\} \\ s &\in \{0.1, 0.2, 0.3 \dots 1\} \\ R_{wc} &= \left\{ \begin{array}{c} 1 \; if \; \mathcal{T}_L < \mathbf{o}_i \leq \mathcal{T}_U \\ 0 \; otherwise \end{array} \right\} \end{aligned} \quad (1)$$

Classifiers can work on the x or y co-ordinates of either the dominant or non-dominant hand. Each classifier covers a strip of a given constant width, either in the x or y plane. Boosting is used to combine these weak classifier strips, to create areas relative to the signer as shown in Figure 1. The strips are shown by increasing the luminosity of the pixels. When many weak classifiers overlap, the area turns white. As can be seen, the white areas coincide with the area being learnt, *i.e.* Figure 1(a) 'face' and Figure 1(b) 'upper arm'.

2.2. *Sig* Classifiers

Sig sub-units describe the motion of a sign and require classifiers which encode temporal information. The tracking



(a) Upper Arm (b) Face

Figure 1: Examples of tracked *Tab* classifiers for the areas 'upper arm' and 'face'. Boosting combines strips in the x and y planes to show where the hand is expected to be for each *Tab* label. The lighter the area in the picture the more strips are overlaying it.

provides a frame by frame set of co-ordinates for the hands so motions can be described by changes in these values. The sub-units from BSL linguistics do not encode magnitude information. Therefore the classifiers used to describe them need to encode non-magnitude dependant information. If the values from the tracking are concatenated temporally into 2D vectors, then it is possible to examine individual components across time. In this way, a weak classifier can look for changes in, for example, the x co-ordinate of the dominant hand. This would encode left and right motion of the dominant hand. Component values can either increase, decrease or remain the same, from one frame to the next. If an increase is described as a 1 and a decrease or 'no change' is described as a 0 then a Binary Pattern (BP) can be used to encode a series of increases/decreases. A temporal vector is said to match the given BP if every '1' accompanies an increase between concurrent frames and every '0' a decrease/'no change'. This is shown in Equation 2 where $\mathbf{O}_{i,t}$ is the value of the component, $\mathbf{o}_i$, at time t and $\mathbf{bp}_t$ is the value of the BP at frame t. See Figure 2 for an example where feature vector A makes the weak classifier fire, whereas feature vector B fails, due to the ringed gradients being incompatible.

$$\begin{aligned} R_{wc} &= |\max_{\forall t}(BP(\mathbf{O}_{i,t})) - 1| \\ BP(\mathbf{O}_{i,t}) &= \mathbf{bp}_t - d(\mathbf{O}_{i,t}, \mathbf{O}_{i,t+1}) \\ d(\mathbf{O}_{i,t}, \mathbf{O}_{i,t+1}) &= \left\{ \begin{array}{c} 0 \; if \; \mathbf{O}_{i,t} \leq \mathbf{O}_{i,t+1} \\ 1 \; otherwise \end{array} \right\} \end{aligned} \quad (2)$$

Discarding all magnitude information would mean that salient information might be removed. To retain this information, boosting is given the option of using additive classifiers as well. These look at the average magnitude of a component over time. The weak classifiers are created by applying a threshold, $\mathcal{T}_{wc}$, to the summation of a given component, over several frames. This threshold is optimised across the training data during the boosting phase. For an additive classifier of size T, over component $\mathbf{o}_i$, the response of the classifier, R_{wc}, can be described as in Equation 3.

$$R_{wc} = \left\{ \begin{array}{c} 1 \; if \; \mathcal{T}_{wc} \leq \sum_{t=0}^{T} \mathbf{O}_{i,t} \\ 0 \; otherwise \end{array} \right\} \quad (3)$$

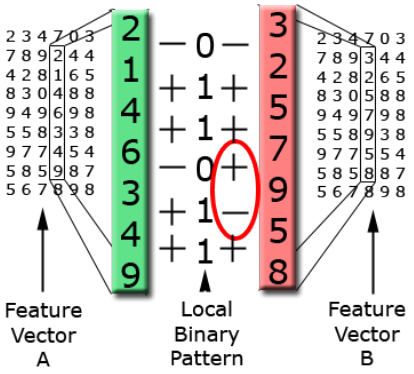


Figure 2: An example of a BP being used to classify two examples. A comparison is made between the elements of the weak classifiers BP and the temporal vector of the component being assessed. If every '1' in the BP aligns with an increase in the component and every '0' aligns with a decrease or 'no change' then the component vector is said to match (*e.g.* case A). However if there are inconsistencies as ringed in case B then the weak classifier will not fire.

Boosting is given all possible combinations of BPs, acting on each of the possible tracking components. The BPs are limited in size to being between 2 and 5 changes (3 - 6 frames) long. The additive features are also applied to all the possible components, but the lengths permitted are between 1 and 26 frames. Both sets of weak classifiers can be temporally offset from the beginning of an example, by any distance up to the maximum distance of 26 frames.

2.3. *Ha* Classifiers

Ha sub-units explain the hand arrangement present in a sign, *e.g.* which hand is higher or whether they are interlinked. Using the tracked positions on each frame, the x and y values of all points can be compared. This can be done using a magnitude comparison, as illustrated in Equation 4 where $\mathbf{O}_{i,t}$ is the first component and $\mathbf{O}_{j,t}$ is the second, both on frame t. Though this does not encode any information about the magnitude of the difference required for the weak classifier to fire. Alternatively, for each point-comparison, 11 weak classifiers are built. Each requiring a different magnitude difference to fire. The difference magnitude, $\mathcal{T}_{wc}$, is selected from a set of 0 to 50 pixels in 5 pixel steps as shown in Equation 5. This selection of thresholds gives $(36!/(34!*2!))*11 = 6930$ possible weak classifiers.

$$R_{wc} = \left\{ \begin{array}{c} 1 \; if \; (\mathbf{O}_{i,t} \leq \mathbf{O}_{j,t}) \\ 0 \; otherwise \end{array} \right\} \quad (4)$$

$$\begin{aligned} R_{wc} &= \left\{ \begin{array}{c} 1 \; if \; \mathcal{T}_{wc} \leq (\mathbf{O}_{i,t} - \mathbf{O}_{j,t}) \\ 0 \; otherwise \end{array} \right\} \\ \mathcal{T}_{wc} &= 0, 5, 10 \ldots 50 \end{aligned} \quad (5)$$

3. Data Set

This work uses the same 164 sign data set as Kadir *et al.* (Kadir et al., 2004) but with extra annotation at the sub-unit level. 7410 *Tab* examples, 322 *Ha* examples and 578

Label	*Ha*	*Ha*M
left_up	82.14%	87.21%
right_up	67.83%	93.88%
side_by_side	91.01%	93.85%
contact	81.45%	87.13%
left_nearer	91.67%	100.00%
right_nearer	96.43%	87.50%
interlink	73.68%	96.55%
Mean	**83.46%**	**92.30%**
Std Dev	**5.13pp**	**10.31pp**

Table 1: Results of *Ha* and *Ha*M tracking based classifiers

Sig were hand labelled for training viseme classifiers. The data set consists of 1640 sign examples. Signs were chosen randomly rather than picking specific examples which are known to be easy to separate. *Tab* sub-units are static and happen on a single frame with multiple frames per sign. As such, the example counts are higher than those for *Sig* which are movement visemes and happen across multiple frames. *Ha* visemes are also static, however, they change more quickly within a sign than *Tab* visemes. As a result, there are often only one or two frames per sign which contain the *Ha* value given by the BSL Dictionary.

4. Sub-Unit Results

For the tracked classifiers, six different types of classifiers were tested for the three different sub-unit types. For *Ha* sub-units, there are two possible classifiers; those which make a binary comparison on the x and y positions of the hands, and those described in more detail in Section 2.3. where the magnitude of the difference is taken into account. For *Tab*, the two classifiers tested are based on the labelling, the first uses the labels independently, the second implements the hierarchical structures described in Section 2.1. The *Sig* sub-unit classifiers were tested with both the standard labels and the revised component labels.

Classifiers are trained on sub-units from four out of ten available signs, then tested on the sub-units from the remaining six. The results shown are taken from the diagonals of confusion matrices across each sub-unit type.

4.1. *Ha* Classifiers

First is the comparison between the results of the binary comparison *Ha* classifiers and the comparators which take the magnitude into account shown in Table 1. The former manage a good response with a mean true-positive rate of 83.46% achieving a maximum 96.43%. The classifiers which include magnitude manage better on all labels but one, with a true-positive mean of 92.30%, 9pp better than the previous results. The magnitude comparators also result in a more consistent classifier with a Standard Deviation (Std Dev) half that of the binary comparison classifiers.

4.2. *Tab* Classifiers

Next, the tracked *Tab* classifiers are examined with the original labels, see Table 2, the mean true positive classification rate is poor, achieving only 46.95% with some classifiers getting 0%. Notably where it fails to distinguish between 'upper_arm' and 'lower_arm'. Moving to the hierarchical label system, the first thing to note is that confusions are only considered between labels of the same level (*e.g.* 'face' is compared to 'arm' but not to 'face_lower' or 'arm_upper'). This is because the data for some of the lower levels is used as positive training data for the higher labels, so a direct comparison cannot be made with the non-hierarchical labels due to the changes in the way the confusion matrices need to be constructed. However, when using these labels, in the confusion matrix, the mean true-positive rate is 79.84%, 33pp higher than the non-hierarchical version. There is also a reduction of 10pp in the Std Dev suggesting that this again gives a more consistent classifier.

Label	*Tab*	*Tab*H
arm		97%
chest	80%	35%
face	47%	95%
arm_low	85%	71%
arm_up	0%	54%
chest_right	0%	88%
chest_up	75%	97%
face_low	53%	78%
face_side		75%
face_up	71%	81%
chest_up_shoulder		91%
face_low_mouth	30%	59%
face_low_nose	39%	83%
face_low_underchin	72%	95%
face_side_cheek	30%	67%
face_side_ear	30%	81%
face_up_eyes	25%	75%
chest_up_shoulder_right	69%	98%
Mean	**46.95%**	**79.84%**
Std Dev	**27.90pp**	**17.73pp**

Table 2: Results of *Tab*tracking based classifiers.

4.3. *Sig* Classifiers

The two versions of tracked *Sig* classifiers, like the previous tracked *Tab* classifiers, are based solely on a change in the way the training labels are used. The difference between the *Sig* classifiers and the other sub-unit classifiers, is that the *Sig* classifiers are boosted across more than one frame, so the training data is used not only to create the classifiers but also to choose the length of the chosen strong classifier. Confusion matrices are calculated for each possible length over the training data. Table 3 shows the results from the training and testing. *Sig* classifiers (using the original labels) give a training true-positive rate of 62%, which is substantially higher than the test average of 48% achieved when using the training derived lengths.

The outcome is similar when examining the results for the new component based labels *Sig*C. The best training lengths give an average of 79% which is an increase of 17pp over the non component based training system. This is reflected in the results when using the training lengths on the test data, where a 53% level is attained, a 5pp increase on the previous result.

Label	*Sig*C Train max	*Sig*C Test	*Sig* Train max	*Sig* Test
B_apart	98%	74%	97%	81%
B_circ_tog_down_alt	69%	41%	0%	0%
B_circ_tog_tow	82%	49%	0%	0%
B_down	63%	53%	78%	67%
B_tog	82%	52%	88%	66%
B_tow_away_alt	92%	45%	94%	44%
B_up	91%	71%	80%	72%
B_up_down	100%	93%	93%	87%
B_up_down_alt	100%	97%	0%	0%
D_away	58%	36%	75%	46%
D_away_down	67%	28%	0%	0%
D_circ_left_down	83%	84%	100%	95%
D_circ_left_tow	100%	98%	100%	98%
D_down	44%	40%	77%	74%
D_down_away	46%	20%	0%	0%
D_left	90%	48%	63%	61%
D_left_right	93%	51%	77%	33%
D_right	48%	27%	33%	28%
D_tap	67%	24%	44%	39%
D_tow	87%	26%	100%	51%
D_tow_away	91%	37%	76%	48%
D_wrist_tow_away	81%	58%	64%	34%
D_up	88%	74%	96%	90%
Mean	**79%**	**53%**	**62%**	**48%**
Std Dev	**18%**	**24%**	**38%**	**33%**

Table 3: Results of *Sig* classifiers and *Sig*C classifiers using component based labels. The first column shows the maximum training classification achieved, the second shows the rate when using the length, found via training, on the test data.

5. Sign Level Results

For completeness, basic sign level results are shown using the same Markov Model as that in (Kadir et al., 2004) The second stage classifier is trained on the previously used four training examples plus one other, giving five training examples per sign. Shown in Table 4 as the results of combining the various sub-unit classifiers with the Markov model. The best results are gained using the magnitude comparisons for *Ha*, the hierarchical representation of *Tab* and the basic *Sig* classifiers, getting 63%.

6. Conclusions

Tests were conducted using boosting to learn three types of linguistic sub-unit, which are then combined with a simple second stage classifier to learn word level signs. By basing the sub-units on the linguistic taxonomy there is greater scope for using data and priors from the linguistic domain as well as using the sub-unit classifiers to aid in data annotation. However, this data set is few in repetitions, with only 4 per sign for training the viseme level classifiers. This means that there are not always enough examples to fully separate each viseme type and more information than just the viseme might be encoded by the classifier. It is also lacking in the number of signs it contains, having only 164 signs, which is insufficient to fully represent all the visemes for which classifiers should be learnt. However, there is currently no other publicly-available data set, which has sub-unit labelling at the temporal level, with which to better train the classifiers. It is for this reason that future work should investigate other sources of data whilst continuing to use a sub-sign representation allowing large lexicons to be tackled effectively.

Combination	*Ha* *Tab*H *Sig*	*Ha*M *Tab* *Sig*	*Ha*M *Tab*H *Sig*C	*Ha*M *Tab*H *Sig*
Mean	35.7%	60.6%	55.5%	63.0%
Minimum	33.9%	57.7%	52.7%	61.2%
Maximum	36.6%	62.4%	57.1%	65.1%
Std Dev	0.8	1.6	1.4	1.3

Table 4: Classification performance using sub-unit level classifiers, combined together by a basic Markov Model LUT, trained on five examples. *Ha* uses binary comparisons between values, whereas *Ha*M uses the magnitude of the difference between values. *Tab* does not use the hierarchical structure of this sub-unit class, *Tab*H includes this structure. *Sig*C uses the component based labels whereas *Sig* use the standard labels.

7. References

British Deaf Association. 1992. *Dictionary of British Sign Language/English*. Faber and Faber.

P. Buehler, M. Everingham, D. P. Huttenlocher, and A Zisserman. 2008. Long term arm and hand tracking for continuous sign language TV broadcasts. In *Procs. of BMVC*, pages 1105 – 1114, Leeds, UK, September 1 – 4.

Yoav Freund and Robert E Schapire. 1995. A decision-theoretic generalization of on-line learning and an application to boosting. In *Procs. of European Conference on Computational Learning Theory*, pages 23 – 37, Barcelona, Spain, March 13 – 15. Springer-Verlag.

J.W. Han, G. Awad, and A Sutherland. 2009. Modelling and segmenting subunits for sign language recognition based on hand motion analysis. *Pattern Recognition Letters*, 30(6):623 – 633, April.

T. Kadir, R. Bowden, E.J. Ong, and A Zisserman. 2004. Minimal training, large lexicon, unconstrained sign language recognition. In *Procs. of BMVC*, volume 2, pages 939 – 948, Kingston, UK, September 7 – 9.

P. Viola and M Jones. 2001. Rapid object detection using a boosted cascade of simple features. In *Procs. of CVPR*, volume 1, pages 511 – 518, Kauai, HI, USA, December.

Pei Yin, T. Starner, H. Hamilton, I. Essa, and J. M Rehg. 2009. Learning the basic units in american sign language using discriminative segmental feature selection. In *Procs. of ASSP*, pages 4757 – 4760, Taipei, Taiwan, April 19 – 24.