Digital tools for sign language research: towards recognition and comparison of lexical signs

Fragkiadakis, M.

CHAPTER 4

Sign and Search: Sign Search Functionality for Sign Language Lexica

Sign language lexica are a useful resource for researchers and people learning sign languages. Typically, users search for signs using unique identifiers or primary features like handshape and location. This study introduces a reverse search functionality, allowing users to perform a sign in front of a webcam to find matching signs. Using OpenPose for pose estimation, we evaluated four techniques (PCA, UMAP, DTW, Siamese Networks) to measure distances between 20 query signs performed by eight participants against a 1200 sign lexicon. The results indicate that UMAP accurately predicts a matching sign with 95% and 80% accuracy at the top-50 and top-20 levels, respectively, using dominant hand arm movement. DTW showed an average accuracy of 70% at the top-20 level. Enhancing the lexicon with more sign instances increased accuracy to 90% at the top-10 level with DTW. Our method is applicable to any sign language lexicon, regardless of size, and can measure variations in signing across different signers or languages.[1]

[1]Chapter based on: Manolis Fragkiadakis and Peter van der Putten. "Sign and Search: Sign Search Functionality for Sign Language Lexica". In: Proceedings of the 1st International Workshop on Automatic Translation for Signed and Spoken

4.1 Introduction

Sign language lexica are a valuable source for people learning sign languages, teachers and parents who need to communicate in signs with their deaf children as well as researchers studying the languages in question. To their core, these lexica allow the user to submit a query containing a unique identifier that by definition refers to a sign (commonly referred to as gloss) and retrieve a video or an image of that sign. In addition to this functionality, some lexica let the user define the formal parameters of the target sign (i.e. its location, handshape, or movement) and retrieve all the signs that contain these features. It is then at the users' discretion to view all the provided signs and select the desired ones. These search functionalities are particularly useful as sign languages, contrary to spoken languages, do not have any unified notation system for sign representation.

Even though a sign search functionality which is based on formal parameters is a user-friendly option in sign language lexica, it still requires human intervention. Dictionary compilers have to manually link these values to the individual videos of signs. This is a time consuming task and, as Zwitserlood [184] discusses, it is the reason why only a few of such dictionaries exist to date. More importantly, according to Zwitserlood, these dictionaries are unidirectional "giving only signed translations of words from a spoken language in a one-to-one relation" [184]. Furthermore, as the retrieved results contain only the parameters selected by the user, the signs are presented in no particular order.

In this chapter, we describe a methodology and its experimental results for multi-directional search functionality for sign language lexica. Our proposed method, extending on previous efforts by Schneider et al. [144] and Fragkiadakis et al. [61], utilizes either the Uniform Manifold Approximation and Projection for Dimension Reduction (UMAP) technique or the Dynamic Time Warping (DTW) algorithm to measure the distance between a query sign and all the signs in a lexicon. Both techniques require no training thus making our methodology applicable in any sign language. These methods have been compared to two other techniques namely Principal Component Analysis (PCA), Euclidean distance.

Languages (AT4SSL). Virtual: Association for Machine Translation in the Americas, 2021, pp. 23–32.

The chapter is organized as follows: in Section 4.2 we give an overview of studies focused on isolated sign recognition. In addition, we discuss research which has been conducted in relation to search functionality for sign language lexica or has the potential to be applied in that domain. In Section 4.3 we describe our methodology regarding the extraction of the body joint coordinates as well as the methods and algorithms compared in this study. In Section 4.4 we present the results of our experiments. We discuss them in Section 4.5 and conclude and motivate future research in Section 4.6.

4.2 Related Work

The following section presents the related work regarding the isolated sign recognition and the sign search functionality domains.

4.2.1 Isolated Sign Recognition

Compared to sign language recognition where a sign needs to be predicted from a continuous stream of signs, isolated sign recognition predicts one sign at a time. As a result, studies focused on such a task tend to vary in their suggested implementations from continuous sign language recognition research which typically needs to detect additional sign boundaries. In this study, we focus on isolated signs as such trimmed data are to be expected when a person is looking for one sign in a dictionary.

In the domain of isolated sign recognition, a fair amount of studies have focused on recognizing letters and finger-spelling and not full signs [133, 93, 95, 111, 110, 129]. Pugeault and Bowden [133] have used random forests to predict hand-shapes corresponding to letters of American Sign Language using appearance and depth images. Keskin et al. [93] reported an accuracy of 97.8% on the same data-set using a multi–layered randomized decision forest (RDF) framework for articulated hand pose estimation. Kirac et al. [95] used random forests for regression (RDF-R) to estimate the 3D hand-shape and their implementation was shown to outperform previous methods.

Marin et al. [111] have combined the data from the Kinect and Leapmotion sensors and reported an accuracy of 91.3% on a subset of ten ASL letters. In a later study [110], they used a multi-class support

vector machines (SVMs) and random forest (RF) and reported a better accuracy of 96.50% for ten ASL letters.

In parallel to the finger-spelling recognition studies, various approaches have been proposed for isolated sign recognition. Similarly though to the finger-spelling recognition domain a remarkable amount of studies have used some kind of depth sensory system (most notable Kinect) to capture and perform classification tasks. Lang et al. [103] proposed a framework that makes use of Kinect and is able to recognize 25 signs from German Sign Language (DGS) with an accuracy of 97%. Almeida et al. [118] presented a methodology for feature extraction in Brazilian Sign Language using Kinect and achieved 80% accuracy on 34 signs. Mehrotra et al. [114] recognized 37 Indian Sign Language signs and achieved an accuracy of 86.16% using Support Vector Machines (SVMs). Kumar et al. [102] have reported an accuracy of 96.3% on 50 signs by combining Hidden Markov Models (HMM) and Bidirectional Long-Short-Term Memory (BLSTM) neural networks as well as Kinect and Leapmotion sensors.

Various systems for isolated sign recognition have been developed with 2D computer vision techniques. Nandy et al. [122] recognized 22 ISL signs with up to 100% accuracy using the K-nearest neighbor algorithm. Ahmed and Aly [6] used Hidden Markov Models (HMM) to predict 23 isolated signer dependent Arabic Sign Language signs using feature vectors resulted from Local Binary Patterns (LBP) and Principal Component Analysis (PCA). More recently, Ibrahim et al. [80] have reported an accuracy of 97% on recognizing 30 isolated signer-independent Arabic Sign Language signs using a skin-blob tracking technique to recognize and track the hands.

Major disadvantages of all the previous methods reported are: the use of depth sensors and the limited vocabulary tested. These drawbacks make their applicability limited for real use in sign language lexica as most likely the accuracy will degrade in a larger vocabulary tested and most importantly will require special equipment (such as a Kinect sensor) from the users.

Recenly, Bilge et al. [14] proposed a framework for zero-shot learning that uses hand and full body regions along with a combination of 3D-CNNs and LSTMs. These types of neural networks have been previously described in Chapter 2. Their results suggested a top-1 accuracy of 20.9% and top-5 accuracy of 51.4% on a 250 class ASL data-set. Li et al. [104] presented a large-scale Word-Level American

Sign Language (WLASL) video data-set which contains more than 2000 signs performed by over 100 signers. They have additionally compared different deep learning models for word-level recognition and their results showed that pose-based and appearance-based models achieve a top-10 accuracy of 62.63% on 2,000 glosses. Recently, Hosain et al. [78] proposed a pose directed pooling approach to extract additional features from 3D ConvNet. Their results reached a 84.71% and a 75.71% top-10 accuracy on the 1000 and 2000 signs WLASL data-set respectively.

Sincan and Kelles [147] presented a Turkish Sign Language data-set (AUTSL) which contained 226 signs performed by 43 different signers with both RGB and depth information. They have additionally trained several deep learning models and their results showed an accuracy of approximately 62% in a user-independent data-set. On a different Turkish Sign Language data-set (BosphorusSign22k) Gökçe et al. [67] experimented with training separate deep learning models, each specialized on different body region, and fusing their predictions in a score-level manner. Using the data of 5 native signers, each performing 744 signs, they trained the different models and tested them using the data of one signer. Their results suggest that by fusing the hands, body and face data the accuracy reaches its maximum value.

4.2.2 Search Functionality for Sign Language Lexica

Over the last decade, many research projects have examined the use of computer-vision techniques to allow a user to search a sign in a database or lexicon by performing it in front of a camera or sensor. Cooper et al. [39] have used sub-unit features and classifiers to detect motion, sign-location and handshape. Subsequently, these features have been passed to Markov Models to recognize a sign. Their results showed that the correct sign appeared at the top-1 retrieved signs with an accuracy of 73% and at the top-10 retrieved signs at 86.9%. However, their study used the data of only one signer and required prior annotation of the sub-unit features as well as training of the different classifiers.

Elliott et al. [50] developed a look-up tool for signs using Microsoft's Kinect sensor. In their study, the motion of the hands, as well as their location, have been extracted and used to create a binary feature vector which was used as an input to a sign level classifier for recognition. Their results reported an accuracy of 95% at the top-4 retrieved signs on a dictionary of 20 signs and 85.1% at the top-4 signs on a 40 signs

dictionary.

A year later, Wang et al. [168] have created a semi-automatic search functionality. In their system, a user marks the start and end frames of a sign and denotes whether the sign is one- or two-handed. Consequently, the system detects the hands on the basis of skin color and motion. The user can correct, if needed, the detected hand locations and pass the query to the system. Using Dynamic Time Warping their approach computes the similarity between the query sign and all the signs in the database. Their results suggest a 78% accuracy on the top-10 retrieved signs on a 1113 sign lexicon. While the accuracy rate is high enough, the user still needs to indicate the handedness feature (one- or two-handed) as well as the duration of the sign. Additionally, the data-set used in this study has been recorded under studio conditions posing the question of applicability on noisy real-life conditions in the video query.

Conly et al. [37] have used the same data-set and Dynamic Time Warping to match a sign on an American Sign Language dictionary. Using Microsoft's Kinect they detect the hand positions and perform sign matching. Their results suggest an accuracy of 77.3% on the top-50 retrieved signs. A major advantage over Wang's et al. [168] implementation is that this system does not require the intervention of the user.

Metaxas et al. [115] have developed a framework that analyzes handshape, orientation, location, and motion trajectories to recognize 350 ASL signs. By passing the extracted features into Hidden Conditional Ordinal Random Fields (HCORF) they achieve a top-1 accuracy of 93.3% and a top-5 accuracy of 97.9%.

Vidalon and de Martino [176] have created a system for Brazilian Sign Language recognition using DTW, Nearest-Neighbor classifier and Kinect. On a data-set of 107 signs, they have reported an accuracy of approximately 98%. A major drawback of their results is the fact that their data-set is user-dependent.

As discussed earlier, the majority of the aforementioned studies use either a depth sensor or computer-vision techniques. These techniques primarily rely on color and motion detection algorithms, as feature extraction methods, which imposes additional problems as discussed in Chapter 1. Such techniques can be prone to errors and most importantly require studio conditions in order to predict the required features such as the face and the hands. While it's true that videos in a lexicon can be under controlled conditions, the same cannot be assumed for the query

videos. These can come from any conceivable environment and lighting situation, hence the need for the capturing technique to cater to the widest possible range of scenarios.

In 2017 Cao et al. [27] presented a framework for multi-person 2D pose estimation, OpenPose. This framework can efficiently detect body, foot, hand and facial key-points from a simple RGB video or picture. Its high accuracy, performance and easy implementation make it the ideal framework to parse sign language and gestural videos. Its output consists of multiple json (or other formatted) files containing all the pixel x, y coordinates of the body, hand and face joints. Most studies use OpenPose to pre-process the videos and use its output to further train or compare machine and deep learning models.

Schneider et al. [144] have used OpenPose as well as DTW and Nearest-Neighbor algorithm to perform classification on six gestures. Their results suggested an accuracy of 77.4%. Most recently, Fragkiadakis et al. [61] have used OpenPose and DTW to predict a sign recorded using a webcam from a 100 sign lexicon. Their method predicted the matching sign with an 87% and 74% accuracy at the top-10 and top-5 retrieved signs by using the path of the dominant hand's wrist.

This study extends on previous efforts for efficient sign ranking for sign language lexica by:

- Considering a far larger lexicon compared to previous efforts: 1200 signs in total
- Comparing four different algorithms: Siamese neural networks (SNN), Principal Component Analysis (PCA), Uniform Manifold Approximation and Projection for Dimension Reduction (UMAP) and Dynamic Time Warping (DTW)
- Comparing three different skeletal joint combinations (upper body, dominant hand arm, dominant hand wrist)
- Exploring potential accuracy increase by adding more sign instances in the lexicon

An important difference from previous studies in search functionality for sign language dictionaries is that in our case we expect signers to not "properly" sign a particular sign. As Alonzo et al. [9] discuss, it is possible that people would not remember exactly how a sign is

performed, and as a result, they might sign it slightly differently. Thus it is expected that the matching sign would not be in the first retrieved sign result. This is precisely the reason why we tested our methodologies on a data-set that contains signs performed also by people with no or little experience in sign language. In most sign language data-sets used for sign language recognition tasks, signs are mostly performed by people familiar with sign languages. However, sign language lexica are intended also for people with little knowledge of sign language. As a result, high variability is expected when recording a sign.

Another limitation posed in the current study is that sign language lexica do not often contain multiple instances of a particular sign. While various studies using deep learning techniques have shown high accuracy in predicting different signs [104, 78, 147, 67], they cannot be used in our case. These techniques often require vast amount of data in order to be trained which might not be available on all sign language lexica. That is the reason why one of the algorithms tested in this study is a technique within the deep learning domain specifically designed for data-sets with few examples per class (i.e. sign). Our main goal is to develop a system that can be easily used in any sign language lexicon regardless of the amount of data in it and most importantly the language itself. However, in this study, we explore the possibility of having a few additional sign instances in the lexicon and their potential benefit to successful sign retrieval.

4.3 Data-sets and Methods

In this section we describe the data prepossessing as well as data-sets and methods used in this study.

4.3.1 Data Pre-processing and Normalization

OpenPose generates x, y pixel coordinates that represent each anticipated body and finger joint's position. These coordinates are tied to the frame size, necessitating normalization to accommodate for various positions within the frame. Given the expectation that all individuals in the dataset (participants and those within the lexicon) would be standing upright before the camera, there's no need to consider rotational invariance.

Rotational invariance essentially means that the recognition or detection of a feature or object remains consistent, regardless of its orientation. In other words, the feature or object can be rotated to any angle, but the algorithm will still be able to recognize it. However, in this case, the assumption that all individuals are standing upright in front of the camera eliminates the need for rotational invariance, as there won't be substantial changes in orientation to account for.

The normalization process is the following: for each detected person in a frame, the neck key-point coordinates are subtracted from all the other key-points. Subsequently, all key point coordinates are being divided by the distance between the left and right shoulder key-point. Finally, a horizontal flip is applied when a participant is left-handed by calculating the average velocity of each hand's wrist. The overall normalization process is based on previous studies by Celebi et al. [29], Schneider et al. [144] and Fragkiadakis et al [61]. Furthermore, all signs have been re-interpolated to 86 frames which is the mean sign length. This is an important step as the input of the siamese networks relies on equal length time series. Additionally, although it makes little difference to DTW's accuracy, equal length inputs make it easier to handle.

4.3.2 Data-sets

For this study we used the Ghanaian Sign Language lexicon (GSL) [60, 73]. This lexicon consists of 1200 signs from one signer and has been compiled for educational purposes to be used in a mobile application. A lot of studies in the isolated sign language recognition field, as seen in section 4.2, have used sign language data-sets from well documented sign languages with primarily signers with light skin tones. We have decided to apply our methodology in a sign language less documented and analyzed with computer vision and machine learning algorithms in order to further explore how these techniques can perform in such conditions.

In addition, the data gathered by Fragkiadakis et al. [61] have been used to compare the different algorithms described in the next section. This data-set contains the data of ten participants. Each one of them performed the same 20 signs, from the original lexicon, in front of a webcam. The data of two participants have been discarded due to inconsistencies of OpenPose on recognizing their right-hand finger's joints and left arm joints.

We have decided to include in the lexicon the data from a random participant every time we tested the methodology. As the lighting conditions on the participants' videos were of poor quality, the predicted body joints had substantially more noise compared to the ones predicted on the lexicon data. By extending the lexicon with another participant's data, we introduced some noise to the otherwise non-noisy data-set. As a result, each participant's sign was compared with 1220 signs in our database (1200 from the GSL lexicon and 20 from another random participant). A complete overview of the participants' data-set and the apparatus used to gather the data can be found in [61].

One of the main goals of this study is to find if and how different skeletal joints affect the accuracy of the algorithms. As a result, we have compiled 3 different data-sets per condition per participant's data. The first data-set contains the upper body joints as well as the dominant hand fingers joints' coordinates resulting in a $86 \times 29 \times 2$ (frames *by* skeletal joints *by* x, y coordinates) dimensionality per sign. Consecutively, the second data-set contains the dominant hand arm joints' coordinates (nose, neck, shoulder, elbow, wrist) resulting in a $86 \times 5 \times 2$ dimensionality per sign. Finally, the data-set regarding the dominant hand wrist data has a 86×2 dimensionality per sign.

4.3.3 Methods

The following section describes the methods and techniques used in this study. Overall, three different algorithms have been used and compared.

Dimensionality Reduction

As described in the previous section, each sign in each compiled data-set can be seen as a multidimensional vector. To properly project it into 2D space while still retaining most of the original information, we used two dimensionality reduction techniques.

The first technique applied is Principal Component Analysis (PCA). PCA is an orthogonal linear transformation that converts the data to a new frame of reference. PCA constructs Principal Components as linear combinations of the initial variables. These components are not correlated and most of the information within the introductory variables is compressed into the first components. By disposing the components with low information and taking into account the remaining components

as new variables, it allows for dimensionality reduction without loosing information. As a technique it has been widely used in the gestural as well as the sign language domain either as a visualization technique or as a pre-processing stage prior to other machine and deep learning stages [71, 142, 75, 64].

Furthermore, the Uniform Manifold Approximation and Projection for Dimension Reduction (UMAP) technique has been utilized. This method has been used instead of another popular dimensionality reduction technique called t-distributed stochastic neighbor embedding (t-sne) [162]. T-sne's inability to preserve the global structure of the data makes it unusable if distances between different clusters or points need to be calculated such as in our case [112]. In contrast, UMAP can better preserve both local and most of the global structure in the data allowing the calculation of distance metrics between clusters. Moreover, the lack of normalization in UMAP effectively reduces the time of computation of the high-dimensional graph.

In our study both PCA and UMAP have been used to reduce the dimensionality of each sign to a single x, y coordinate. Subsequently, we measured the euclidean distance between all the signs of the lexicon and the participants' signs. Accuracy for each participant's sign was measured based on whether the target sign was on the top-n shortest distant signs.

Furthermore, in order to validate the results produced by the UMAP algorithm in its ability to preserve the global distances of the data, we calculated their euclidean distances in the original high-dimensional space. This method has been used as a benchmark to compare the results of both PCA and UMAP.

Dynamic Time warping

In addition to the dimensionality reduction techniques described above, Dynamic Time Warping (DTW) has been used to measure the similarity between the different signs. Dynamic Time Warping is a dynamic programming based time series comparison algorithm to produce a distance metric between two inputs. It has been widely used in the speech recognition domain [1, 11, 121] as well as the gestural and sign language recognition fields as shown in Section 4.2.

In this study we utilize a DTW python implementation with open beginning and ending attributes by [66, 159] which in a preliminary

experiment produced better results compared to the previous DTW implementation by Fragkiadakis et al. [61]. Similarly, we used a median filter with radius $r = 3$ for smoothing the time series signals from the body joints.

Siamese Networks

While training a deep learning model deviates from our primary goal to develop a functionality that relies on unsupervised methods, we believe that comparing these methods with a trained model would be useful for benchmarking. While training a feed-forward deep learning model would need a large amount of data we opted for a different architecture, such as siamese networks, to fit our needs which requires far less data. The following section describes these networks, their configuration and training process.

Siamese neural networks (SNN) are a type of artificial neural network that contains two sub-networks with the same configuration. These networks were first introduced by [20, 36] in order to be applied to signature verification and face recognition tasks. Their primary use is to find the similarity of two inputs by comparing their feature vectors. Typically, they contain two feed-forward neural networks with shared weights that have as an input two vectors. During training the output space gets structured in such a way that the distance between two sample outputs expresses a a semantic similarity.

Due to the different skeletal joints' combinations tested in this experiment, it was necessary to develop and train three different siamese networks based on the different input dimensionalities as described in Section 4.3.2. The first network trained on the upper body data consisted of three convolutional layers with the Rectified Linear Unit (ReLU) activation function and kernel sizes of 7, 5 and 3 respectively. The second network trained on the dominant hand arm's data consisted of three convolutional layers with ReLU activation functions and kernel sizes of 3, 2 and 1 respectively. The last Siamese Network trained on the dominant hand wrist's data consisted of 1 convolutional layer with kernel size of 10 and activation function ReLU followed by two linear layers of 100 and 10 output sizes respectively. The last layer in all siamese networks consisted of a linear layer with input size of 100 and output of 1 reflecting the similarity score.

All developed siamese networks were trained using the contrastive

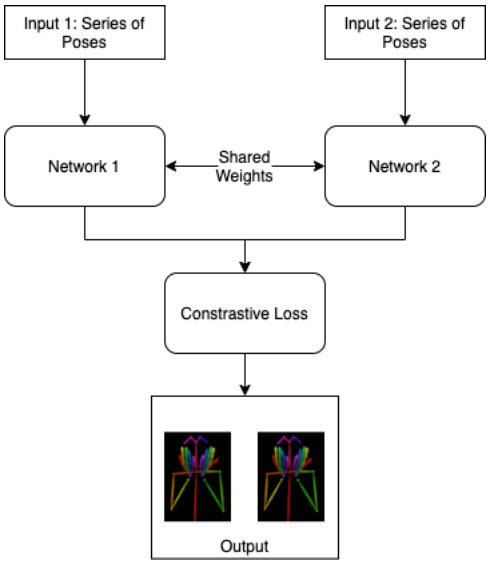
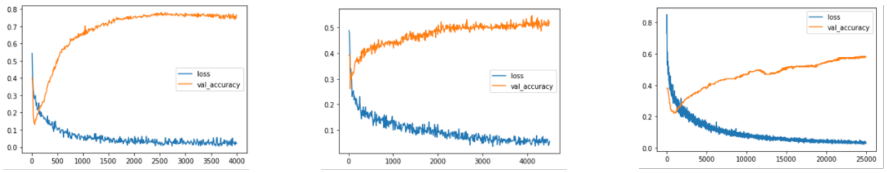


Figure 4.1: Basic structure of the siamese networks trained in this study.

loss function. This function is a distance-based loss function as opposed to conventional error-prediction losses. Its use is to learn the embeddings in which two similar points have a low Euclidean distance and two dissimilar points have a large Euclidean distance.

Siamese networks get trained on combinations of data that have been tagged as similar (1) or different (0). To create such combinations, we have used the data of four random participants (half of the "population"). The data from each sign from each participant have been combined with the data of the same sign from another participant creating a "similar" signs pair. Consecutively, the same sign has been paired with another random sign from another participant creating the "different" signs pair. A basic outline of a Siamese Network can be seen in figure 4.1.

Since these combinations resulted in small data-sets an additional data augmentation technique has been applied. For each "similar" and "different" signs pair a random rotation of $\pm$ 10 degrees has been applied on the skeletal data on one of the signs. This whole process resulted in a total of 1228 signs combinations. Finally, for each skeletal condition (upper body, dominant hand arm, dominant hand wrist) a different data-set containing only the appropriate pose data has been compiled.



(a) upper body (b) right hand (c) wrist

Figure 4.2: Validation accuracy and loss for the different Siamese networks per skeletal condition

Accuracy of these Siamese networks was measured by calculating the absolute value between 1 minus the distance predicted by the network. As such, closer to 0 would mean accurate prediction of similar signs. Each compiled data-set was split to a training and validation set. Validation accuracy and loss during training on each of the Siamese networks based on the different skeletal conditions can be seen in figure 4.2.

Finally, the overall pipeline of the experiment can be seen in figure 4.3.

How many signs?

Some sign language lexica allow their users to submit their own versions of signs. As a result, different instances of the same sign can be stored in the database. One of our research questions is whether having multiple instances of each sign can potentially improve the accuracy of the algorithms. To verify that, we progressively added the 20 signs from the other participants to the lexicon. Subsequently, we measured the average top-1 and top-10 accuracy for each algorithm and each skeletal condition.

Such information can be useful to sign language lexicographers when compiling sign language lexica. They can take advantage of crowd-sourcing material, contributing not only to the augmentation of their lexica but also to the accuracy of the models used for enhanced search functionality.

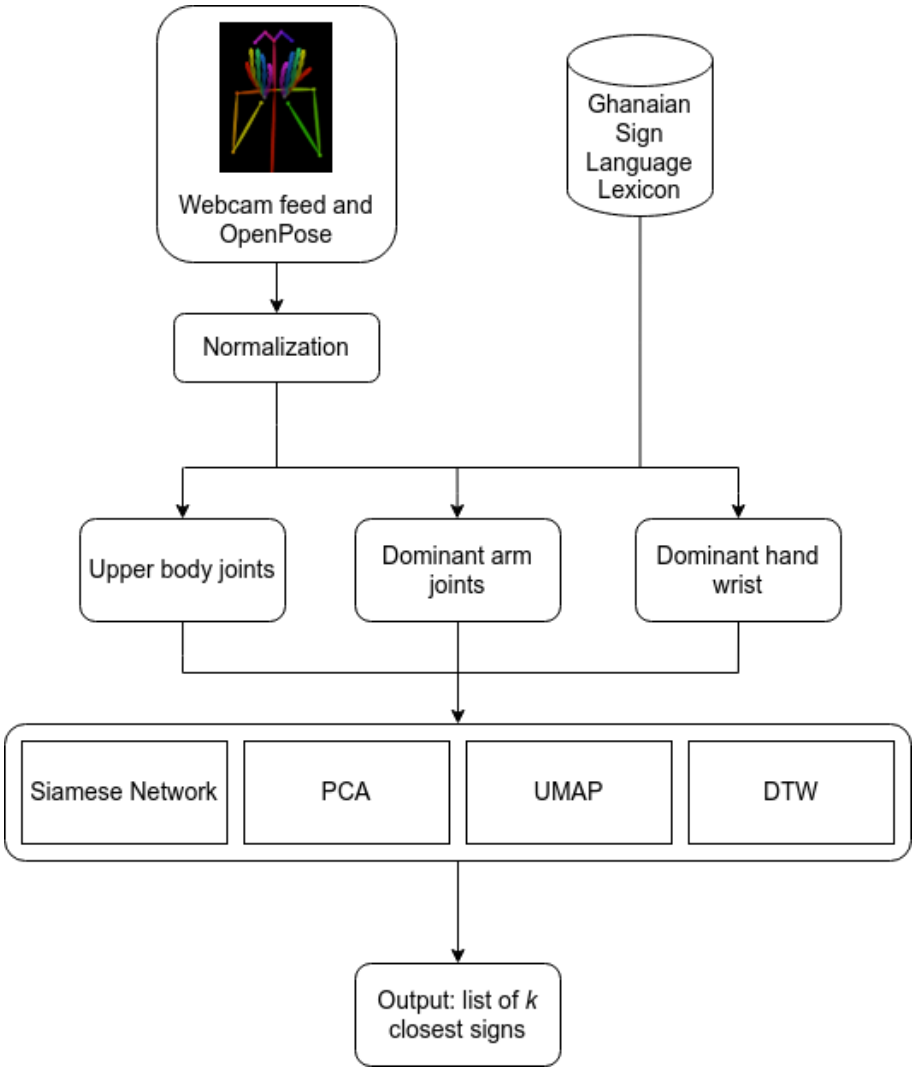


Figure 4.3: Pipeline of the overall study.

4.4 Results

Table 4.1 presents the overall accuracy for each of the skeletal conditions. Top-k refers to the number of signs a user must look up before finding a correct match. Accuracy indicates whether the target sign is present in the top-k retrieved signs and is averaged across all participants and all signs. Additionally, figure 4.4 presents the visualizations of the UMAP algorithm for each of the skeletal condition for one participant. Visualizations of the PCA algorithm can be found in figure 4.5.

Highest accuracy is apparent at a top 50 level at 95% using the UMAP algorithm and the joints of the dominant hand arm. Furthermore, top- 20 rank shows an adequate accuracy at 80% again using UMAP and the dominant hand arm coordinates. The results of the calculated euclidean distances on the original high-dimensional space show an adequate accuracy of approximately 68% at the top-50 rank in both dominant hand arm and wrist data-sets.

Principal component analysis (PCA) performed on average better using the wrist coordinates and showed the highest accuracy at the top-50 at approximately 41%.

DTW showed the highest accuracy at 79% at top-50 rank using the data of the dominant hand wrist and 77% using the dominant hand arm. On average, DTW had the best accuracy at around 70% at the top-20 ranking regardless of the skeletal condition used, with a slight increase noticed using the dominant hand arm data.

Contrary to expectations, using Siamese networks did not yield adequate results. Their highest accuracy was measured at 30% on the top-50 rank using the upper body joints. Using the arm and wrist joints did not produce significant results.

Figure 4.6 presents the top-1 and top-10 accuracy levels using DTW and UMAP that have been computed by incrementally adding participants' data in the lexicon. It can be observed that by adding more sign instances from 6 different participants, the accuracy reached a 90% level at the top-10 retrieved signs using DTW and the upper body and dominant hand wrist data. Furthermore, a raise of approximately 15% can be noticed on DTW using the upper body and dominant hand wrist at the top-1 rank by adding the data of just 2 participants (figure 4.3a). On the other hand, UMAP did not show any significant raise in the top-1 accuracy regardless of the added participants' data and skeletal condition. However, an increase, of approximately 35%, can be seen at

the top-10 ranking level using the data of 2 participants (figure 4.3b).

4.5 Discussion

In this study we have investigated the use of OpenPose and three different implementations as distance metrics for an efficient ranking pipeline to retrieve matching signs from a sign language lexicon. The results demonstrated that on a large vocabulary of 1200 signs, such a task can be achieved with an adequate accuracy rate using the Uniform Manifold Approximation and Projection for Dimension Reduction (UMAP) or Dynamic Time Warping (DTW) and the dominant arm joints' coordinates.

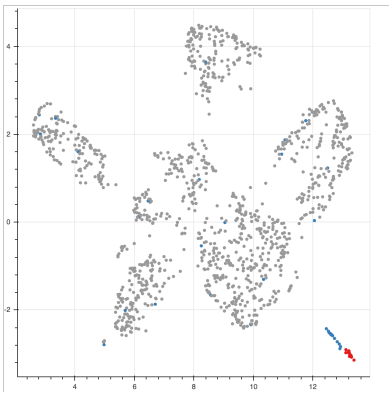
Our results were inadequate considering the use of siamese neural networks for sign search functionality. While validation accuracy during training of these networks reached nearly 80%, testing them on the other participants' data showed poor results. Such behavior can be accounted either on the low amount of data used in the training process, possibly over-fitting the network or on the wrong selected architecture. While during preliminary experiments we have tested different architectures for these networks, the ones that we ultimately chose were the ones used in this study. However, only one skeletal condition (upper body) showed high enough validation accuracy during the training process. The other data-sets using other skeletal data did not even produced adequate validation accuracy. This makes us believe that the overall dimensionality as well as size of the data-sets is not enough to train these neural networks "from scratch". A possible solution to this could be the re-purpose of another deep learning model trained on different data for the task of our study. This technique commonly referred to as transfer learning can be quite effective when little data are available [150]. However, most pre-trained deep learning models accept an input size of at least 224 [2] which is significantly larger than the extracted body joints from OpenPose used in this study. Possible solutions using transfer learning and OpenPose predicted body joints for efficient sign search functionality on large lexica should be further explored on future research.

With regard to the visualizations produced by the UMAP algorithm, a few observations can be made. Firstly, by using the upper body joints

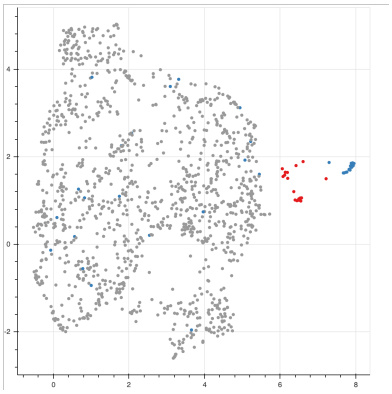
[2] https://pytorch.org/vision/stable/models.html

Skeletal condition	Upper body				Dominant hand arm				Dominant hand wrist			
Top-k	Top - 1	Top - 10	Top - 20	Top - 50	Top - 1	Top - 10	Top - 20	Top - 50	Top - 1	Top - 10	Top - 20	Top - 50
PCA	0.0375	0.1562	0.2187	0.325	0.025	0.1562	0.2125	0.3437	0.0187	0.1562	0.2125	0.4125
UMAP	0.1312	0.4125	0.6562	**0.6937**	0.0812	0.4375	**0.8**	**0.95**	0.1125	0.2562	0.3687	0.4575
DTW	**0.57**	**0.64**	**0.67**	0.687	**0.5188**	**0.65**	0.7188	0.7763	**0.5265**	**0.6562**	**0.7125**	**0.7937**
Euclidean distance	0.2125	0.4625	0.5325	0.6188	0.2625	0.5	0.6063	0.6938	0.1875	0.445	0.55	0.675
Siamese networks	0.0625	0.125	0.1875	0.3	0.1625	0.075	0.1	0.15	0.05	0.1125	0.1375	0.1625

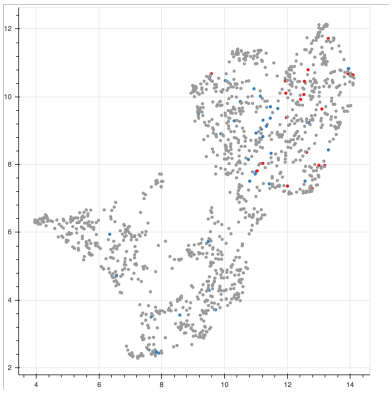
Table 4.1: Sign retrieval accuracy per algorithm (by row) on the three skeletal conditions based on the top-k retrieved signs (highest value per column in **bold**).



(a) upper body

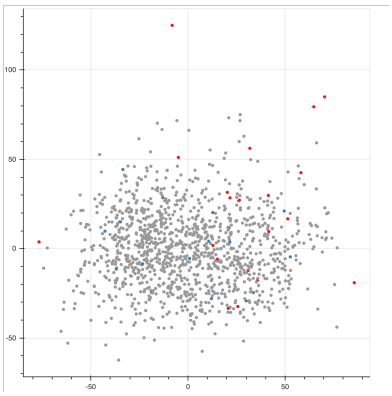


(b) dominant hand arm

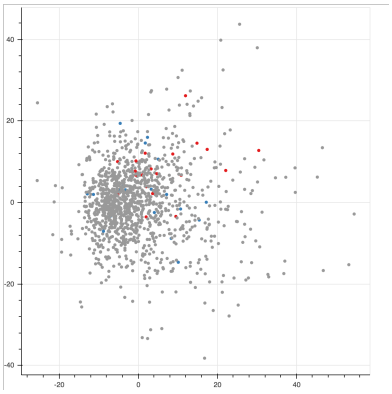


(c) dominant hand wrist

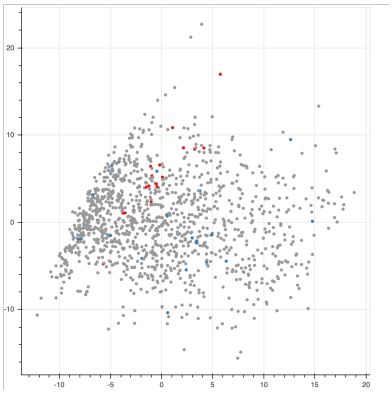
Figure 4.4: UMAP visualizations for one participant for the different skeletal conditions. With red are the signs of the participant and with blue the targeted signs.



(a) upper body

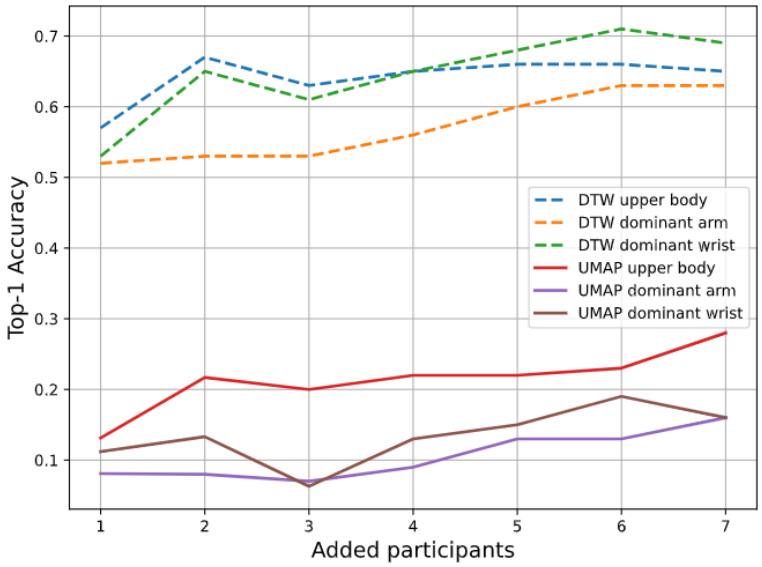


(b) dominant hand arm

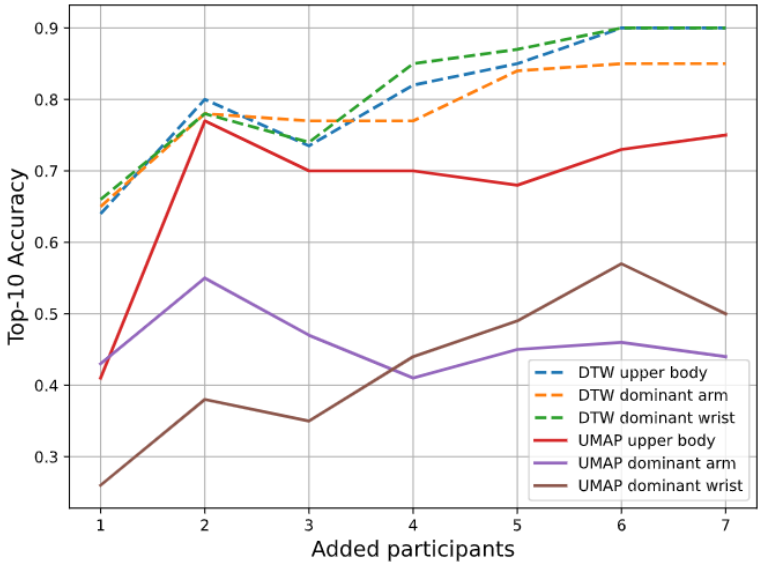


(c) dominant hand wrist

Figure 4.5: UMAP visualizations for one participant for the different skeletal conditions. With red are the signs of the participant and with blue the targeted signs.



(a) Top-1



(b) Top-10

Figure 4.6: Top-1 (a) and Top-10 (b) accuracy using DTW and UMAP based on added participants' data in the lexicon.

UMAP produces discrete clusters. These clusters seem to reflect an abstract representation of the movement of each sign. This behavior has been observed also using the dominant hand arm coordinates, although it is more noticeable using all the upper-body joints. We can observe that signs that have similar movement but different handshapes are grouped close to each other. A hand-picked examples showing some of these signs can be seen in figure 4.7 while figure 4.8 shows the realization of them.

However, special consideration needs to be made when viewing the visualizations produced by UMAP, especially the one using the upper body joints. The distances between the noticeable clusters, as well as their size relative to each other, do not hold any particular meaning. This is because of the use of local distances by the algorithm when constructing the graph. However, our results using the euclidean distances on the high-dimensional space suggest that UMAP preserves the global distances.

Finally, it is worth mentioning that DTW performs equally well irrespective of the skeletal condition used at around 70% at the top-20 rank. Overall, it produces the most stable and consistent accuracy at the top-10 retrieved signs at around 65%. This accuracy level can be further raised reaching even 90% at the top-10 ranking level by adding 6 more sign instances (from different signers) into the original lexicon. This attribute can be further explored by lexicographers by asking users of their lexica to submit their own versions of signs. This process can significantly boost the performance of DTW in its ability of retrieving the closest matching sign. A broad benefit of using such an algorithm is the fact that lexica compilers do not need to re-train any model if more signs or sign instances are added to their lexica.

In general, while our accuracy does not reach the ones reported in Schneider et al. [144] and Fragkiadakis et al. [61] (77.4% top-1 and 74% top-5 accuracy respectively) using similar algorithms and frameworks, our methods have been applied on a far larger lexicon (1200 signs instead of 6 and 100 respectively). As a result, we provide a better approximation on how these methods can actually be used in real world lexica.

4.6 Conclusions

To sum up, we have obtained satisfactory results demonstrating that UMAP and DTW, in combination with the pre-trained pose estimation

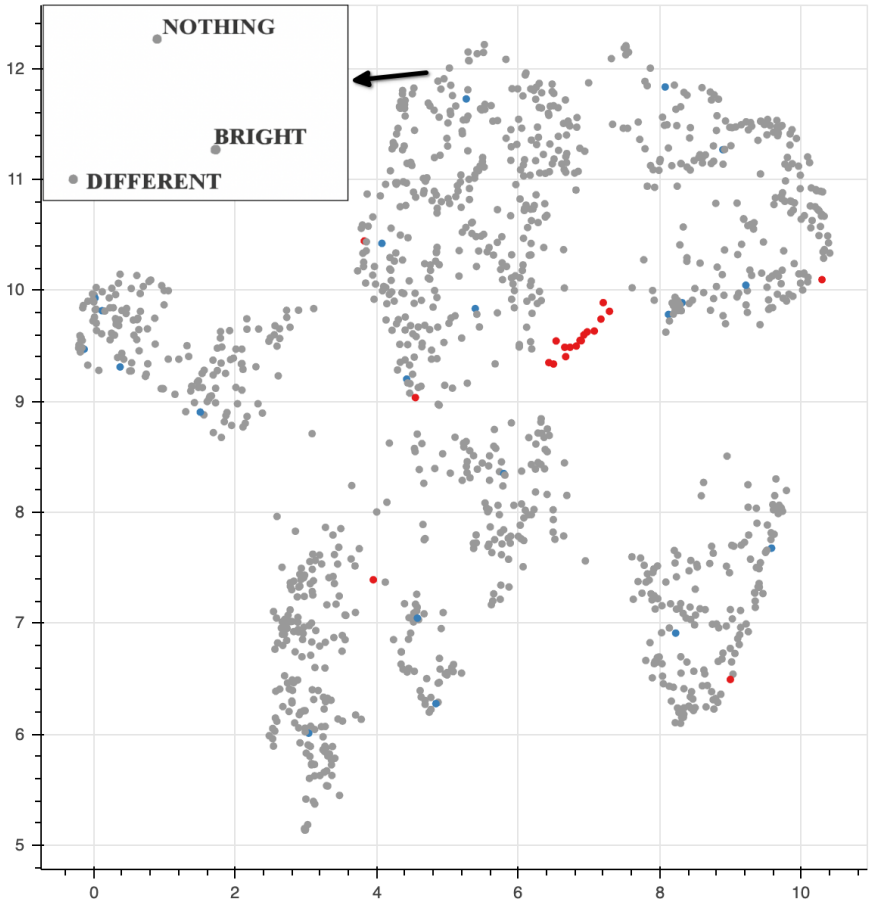


Figure 4.7: UMAP visualization of the upper body joints of one participant and groups of signs with similar movement.

(a) BRIGHT

(b) DIFFERENT

(c) NOTHING

Figure 4.8: Hand-picked examples of signs that were grouped together using UMAP based on the upper body data.

framework OpenPose, can be used as an efficient sign ranking and retrieval system. Our method can effectively be applied to any sign language lexicon without any training process involved.

To the best of our knowledge, this is the first study using UMAP as a dimensionality reduction technique within the sign language domain and showcasing the strength of such an algorithm compared to other implementations.

Future work will focus on exploring additional deep learning implementations for an efficient handshape recognition. Such frameworks could potentially increase the accuracy of the predicted finger joints and rotation resulting in more efficient dimensionality reduction from the UMAP algorithm. In addition, the techniques used in this study will be further explored to measure variation in different sign languages. The results from the use of UMAP and DTW on a large vocabulary suggest that these techniques might be well suited for variation measurement tasks, broadening their use beyond the search functionality for sign language lexica.