KEYWORD SEARCH FOR SIGN LANGUAGE

by

Nazif Can Tamer

B.S., Electrical and Electronics Engineering, Boğaziçi University, 2017

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Electrical and Electronics Engineering
Boğaziçi University

2020

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my thesis supervisor Prof. Murat Saraçlar, not only for his encouragement and endless support during my studies, but also for making research a truly enjoyable experience. Without his motivating and constructive supervision, I would not be able to find the courage for publishing any of our research. I am also grateful to Prof. Lale Akarun for sharing her endless pool of knowledge in the domain of sign language. Without her vision and support, this thesis would not turn into a reality.

The work in this thesis was supported in part by the Scientific and Technological Research Council of Turkey (TUBITAK) under Project 117E059.

ABSTRACT

KEYWORD SEARCH FOR SIGN LANGUAGE

Sign language is the main communication tool for the hearing impaired. However, the information retrieval from sign language is not as easy as in spoken languages. In this thesis, three methods of information retrieval are proposed for sign language. Firstly, a Dynamic Time Warping based Query-by-Example search technique is developed to enable the Deaf to search for information in their native language. Secondly, a translation-based cross-lingual keyword search method is proposed which will enable the people outside of the Deaf community to learn sign language in context with queries from the written language. Finally, as the main contribution of this thesis, a weakly-supervised keyword search technique is designed based on neural word embeddings and attention concept from neural machine translation. This technique is shown to be capable of performing both gloss search and cross-lingual written keyword search; and can be used together with different input features such as human pose estimates and various hand shape features. Our experiments conducted on three datasets, i.e. HospiSign, RWTH-Phoenix-Weather 2014T, and MeineDGS corpus, indicate that using human pose estimates extracted with OpenPose framework generally performs good under different retrieval tasks in sign language, especially when they are combined with Spatio-Temporal Graph Convolution. Furthermore, models based on attention is found to be able to temporally localize the keywords as a by-product of weakly supervised training.

ÖZET

İŞARET DİLİNDE ANAHTAR KELİME ARAMA

İşaret dili, işitme engellilerin temel iletişim aracıdır. Ancak işaret dilinden bilgiye erişim konuşma dilindeki kadar kolaylıkla mümkün olmamaktadır. Bu tezde, işaret dili için üç farklı bilgi erişimi tekniği sunulmuştur. İlk olarak, işitme engellilerin anadillerinde arama yapabilmelerini sağlamak amacıyla Dinamik Zaman Bükmesi temelli bir Örnekle Arama tekniği geliştirilmiştir. İkinci olarak sunulan çeviri tabanlı bir diller-arası arama tekniği ile işaret dili bilmeyenlerin yazı dilinden aramalarla işaretleri bağlam içerisinde öğrenebilmeleri amaçlanmıştır. Son olarak ise, bu tezin temel çıktısı olan düşük güdümlü öğrenmeye dayanan bir anahtar kelime arama tekniği geliştirilmiştir. Hem işaret hem yazı dilinden anahtar kelimelerle arama yapılabilen bu teknik, aynı zamanda iskelet pozisyon bilgileri ve el şekli öznitelikleri gibi farklı özniteliklerle ve farklı dizi kodlama metodlarıyla birlikte kullanılabilmektedir. Bu tez kapsamında HospiSign, RWTH-Phoenix-Weather 2014T ve MeineDGS Corpus veri kümelerinde gerçekleştirilen deneyler, iskelet pozisyonu özniteliklerinin genel olarak farklı işaret dili geri getirimi tekniklerinde, ve özellikle Uzam-Zamansal Çizge Evrişimli Sinir Ağları ile kullanıldığında, iyi başarımlara ulaştığını işaret etmektedir. Bunun yanı sıra, sadece düşük güdümlü öğrenme amacıyla eğitilen dikkat tabanlı modellerin anahtar kelimelerle ilintili zamansal bölgeleri kendiliğinden tespit edebildiği görülmüştür.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

$\tilde{A}$	Adjacency matrix
B	Decoder beam in neural machine translation
$C(q\|B)$	Expected count of a keyword q in the decoder beam B
$\mathbf{C}(i,j)$	Cost matrix in Dynamic Time Warping
c	Context vector
$\mathbf{D}(i,j)$	Cumulative distance matrix in Dynamic Time Warping
$\tilde{D}$	Diagonal matrix
$\vec{d}$	Normalized TF-IDF vector for a context model
H	Pose estimation features
I	Identity matrix
score	Similarity scoring function in attention
r	A translation path in the decoder beam
s_i	Time step i in the encoded sequence
q	Query keyword
W	Weight matrix
$\|V\|$	Number of keywords in the vocabulary
β	Multiplicative parameter in attention score function
γ	Blending ratio in fusion
θ	Additive parameter in attention score function
σ	Neural network activation function
ϕ	Dynamic Time Warping path

LIST OF ACRONYMS/ABBREVIATIONS

AP	Average Precision
ASR	Automatic Speech Recognition
BCE	Binary Cross-Entropy
CLIR	Cross-Lingual Information Retrieval
CNN	Convolutional Neural Network
CTC	Connectionist Temporal Classification
DTW	Dynamic Time Warping
FHD	Fast Hand Descriptors
GRU	Gated Recurrent Unit
HMM	Hidden Markov Model
KWS	Keyword Search
LH	Left Hand
LSTM	Long Short Term Memory
LVCSR	Large Vocabulary Continuous Speech Recognition
MIL	Multiple Instance Learning
mAP	Mean Average Precision
nDGC	Normalized Discounted Cumulative Gain
NMT	Neural Machine Translation
OOV	Out-of-Vocabulary
ReLU	Rectified Linear Unit
RH	Right Hand
RNN	Recurrent Neural Network
SS-DTW	Sub-sequence Dynamic Time Warping
ST-GCN	Spatio-Temporal Graph Convolutional Networks
TF-IDF	Term Frequency Inverse Docuent Frequency
UB	Upper Body
QbE	Query-by-Example

1. INTRODUCTION

Sign language is the visual language of the hearing impaired, and it is used in the communication of the deaf and mute people within themselves as well as in their interaction with others. The information in sign language is mainly carried through a mixed use of visual features such as the signer's hand shapes, body movements, posture, facial expressions, and non-verbal mouthings. This multichannel visual nature of sign language separate it from spoken languages, and also makes automatic sign language recognition a more challenging task.

As with the spoken language, there are great cultural and regional differences between different sign language communities. Moreover, the hearing impaired are usually among the disadvantaged groups in their respective culture. As a direct result of these, most sign languages are underresourced, and the amount of labeled data to be used for automatic sign language recognition systems is scarce even for the most studied sign languages. Furthermore, sign language has a different lexicon, grammar, and word ordering than spoken language; which prohibits the use of spoken language utterance as the ground truth label for sign language video. Due to all these reasons, it is not possible to train retrieval systems in sign language as easily as spoken languages and there is a need for improvement.

The motivation behind this thesis is to develop techniques which make information retrieval from sign language easier. Unlike the people who use spoken languages in their daily lives, the sign language community cannot use easily accessible tools built for written information retrieval. Many of the deaf and mute people learn sign language as their primary mother language, and face hardships in learning the written language. Thus, creating video based query-by-example information retrieval systems help the hearing impaired to reach audiovisual information. Another motivation behind this thesis is to make learning sign language easier for the non-hearing-impaired. Other family members of the hearing impaired usually learn sign language to commu-

nicate with their beloved ones. However, since most sign languages are low resource, the means for learning a sign in context is usually non-existent. The written keyword search methods developed with this thesis can help making it easier to learn sign language.

Keyword search is a sub-problem of content retrieval which aims to search for a written query inside a large and unlabeled utterance. In spoken language recognition, keyword search is studied as a different problem than other retrieval problems such as keyword spotting and term discovery. Content retrieval from continuous sign language, on the other hand, is generally studied together under the umbrella term sign spotting and encompasses query-by-example search, keyword spotting, keyword search, and weakly supervised term discovery. The general approach in sign spotting requires strong supervision during training/learning. Jantunen et al. [2] used dynamic time warping to search for citation form isolated signs in continuous sign language sentences. Yang et al. [3] used conditional random fields to search for 48 in-vocabulary signs that they learned from isolated examples. Ong et al. used hierarchical random fields to search for 48 the signs inside continuous sign language utterances. Although these approaches can obtain good retrieval performances, their search vocabulary is in the order of tens and their real-world applicability to large vocabulary retrieval systems is limited due to the amount of highly-annotated data they require during training.

Another track in sign spotting research is using weakly labeled continuous sign language in both learning time as well as testing. Most of the available sign language data are in the form of sign language interpreting and translations into the written language is the only form of annotation. Since there is no one-to-one relationship between signs and written language words, several studies in the literature focus on discovering signs under the weak supervision of these translations or subtitles. Cooper and Bowden [4] used mining strategies to learn signs by matching the subtitles from TV shows. Farhadi and Forsyth [5] used HMMs to find sign boundaries assuming the sign sequence and the speech transcripts have the same word ordering. Buehler et al. [6] and Kelly et al. [7] applied multiple instance learning (MIL) based strategies

to learn signs from subtitles and Kelly et al. [7] further used the isolated signs they discovered from translations to train a 30-vocabulary sign spotting framework.

1.1. Contributions

In this thesis, various sub-tasks of content retrieval are applied to sign language and different feature extraction strategies and retrieval methods are implemented using only weak labels. The main contributions of this thesis can be summarized under five different topics as follows:

(i) A Dynamic Time Warping (DTW) based Query-by-Example (QbE) sign retrieval method is introduced for sign language and the use of different feature extraction techniques, distance metrics, and trimming strategies are compared for the task of QbE. A national-level conference paper is published on this topic [8].

(ii) A cross-lingual keyword search technique based on neural machine translation is introduced for sign language. Instead of choosing the best translation in a set (beam) of possible translations and searching for keywords in the decoder outputs of a neural machine translation model, a technique where keyword search is performed from expected counts of a word in different translation paths is developed. Keyword search using written language queries is made possible for sign language retrieval and retrieval success under different decoder beam sizes and beam normalization strategies are compared. A national-level conference paper is published on this topic [9].

(iii) As the main contribution of this thesis, a novel attention-based keyword search module is introduced to enable weakly supervised training of sign language retrieval systems. This module is shown to be powerful for both gloss search and cross-lingual search, even in a context where we only have noisy translations as the training labels. Furthermore, it is also shown to be jointly trainable with different video sequence encoding strategies such as spatio-temporal graph convolutional networks, and different hand shape classification methods. The outputs of this

thesis on this topic are one conference paper and two workshop papers which are published in international conferences [10–12].

(iv) A bag-of-words context modeling strategy is introduced for cross-lingual keyword search and shown to be effective in increasing the cross-lingual KWS performance in sign language.

(v) Keyword search performance in sign language is increased by combining pose based and hand shape based keyword search models in a cold fusion approach.

1.2. Thesis Outline

The remainder of this thesis is structured in the following fashion:

- In Chapter 2, related works in the domains of sign language recognition and keyword search are summarized
- In Chapter 3, a query-by-example keyword search method is introduced,
- In Chapter 4, a machine translation based cross-lingual keyword search method is described,
- In Chapter 5, various methods based on a novel keyword search architecture are explained for gloss and cross-lingual keyword search,
- In Chapter 6, experimental setup is provided and empirical results are reported,
- Finally, in Chapter 7, we draw a conclusion on the thesis.

2. RELATED WORK

This thesis focuses on applications of keyword search methods on sign language, which can be classified under the domain of sign spotting. Literature review on sign spotting is given in Chapter 1. Since this is a relatively new research direction, the literature in keyword search applications on sign language, however, is rather limited. In this chapter, related works on closely associated problems and the methodologies we borrowed from other domains are provided for the reader.

2.1. Sign Language Recognition

Sign language recognition is historically the main research field in sign language studies and deals with automatic identification and processing of signed utterances. The two main tracks of sign language recognition are isolated and continuous sign language recognition: Isolated sign language recognition is the task of correctly labeling the sign language video when the sequence is comprised of exactly one sign, thus, it is similar to the broadly studied task of single label action classification. In this setting, both train and test sequences are strongly labeled and thus, learning is easier. Isolated sign language recognition was the main topic of sign language research for a long time, and first studies depended on tracking special hardware that the signers wear, such as colored gloves. Later on, following the general trend in the computer vision domain, isolated sign language recognition has also moved towards working directly from RGB images, where it was often approached by adapting the classification strategies from action recognition community [13, 14], or building gesture recognition systems based on hand shapes [15] and sub-unit discovery [16].

In the more challenging task of continuous sign language recognition, we do not have the start and end temporal annotations for glosses, i.e. the basic units of sign language. Furthermore, unlike spoken language keyword search where we can align speech to phoneme sequences to find the temporal boundaries, gloss annotations in

sign language do not include sub-units analogous to phonemes. Due to these reasons, continuous sign language recognition is often approached as a weakly supervised learning problem. The general procedure in continuous sign language recognition involves two stages: at the first feature extraction stage, video frames are converted into vectors while unrelated features such as signer variance or background are removed. Then, at the second sequential modeling stage, the correspondence between feature sequence and gloss sequence is modeled. Koller et. al. treated output of CNN feature extractors as Bayesian priors to train them together with three-state gloss level HMMs in an expectation-maximization scheme for the sequence modeling of continuous sign language [17]. Using recent advancements in deep learning, Camgoz et al. [18] and Cui et al. [19] trained their Recurrent Neural Network based continuous sign language recognition systems with Connectionist Temporal Classification (CTC) loss, which eliminates the sequence modeling problem by considering all possible alignments between sign language and gloss sequences while calculating the training error. The main assumption in continuous sign language recognition schemes is that we have the signed utterance transcribed into an ordered gloss sequence. Although this makes using alignment based strategies such as HMMs or loss functions such as CTC possible, this level of labeling is usually hard to find for low resource sign languages.

2.2. Keyword Search

Keyword search is generally recognized as a sub-field under the domain of spoken content retrieval, where it is defined as the search for a written query in a spoken corpus. Currently, there are mainly two approaches for keyword search for spoken languages depending on the data availability in the target language: for well-resourced languages such as English and German, the current method for keyword search starts with obtaining possible transcriptions with Large Vocabulary Continuous Speech Recognition (LVCSR) systems. Then, a lattice search is performed along possible transcriptions [20]. Since a language model is used in decoding a spoken utterance, retrieving out-of-vocabulary (OOV) keyword is not possible under this setting. To remedy this, substituting OOV keywords with acoustically-similar proxy words in the

vocabulary [21]. However, it is not possible to apply a similar substitution strategy to sign language since sub-units are not known, and finding phonologically similar glosses from written annotations is not possible.

For low resource languages where the amount of available data is not large enough to solely rely on LVCSR systems for keyword search, pattern matching techniques such as posteriorgram based keyword search are chosen [22]. However, similar to proxy word substitution methods, these methods also rely on the fact that acoustic sub-units are generally conceivable from the written form of the spoken language. However, sign language is not only a low resource language, but also a language where pronunciation of a sign can not be inferred from the written gloss form. Thus, application of such techniques on the keyword search for sign language problem is limited.

Another track which is becoming more popular in the keyword search domain is to employ end-to-end methods while approaching KWS as a weakly supervised problem. Audkhasi et al. [23] was the first to explore end-to-end keyword search with a recurrent neural network based model that summarizes the entire speech utterance in a single vector embedding and compare it with embeddings for different keywords. However, this approach does not enable keywords to focus on relevant parts of the speech utterance during search. In a similar task of keyword spotting, Shan et al. [24] uses attention for temporal focusing while doing single term keyword spotting task. Although attention can enable each keyword to focus on different temporal regions, to our knowledge, this approach was not explored for large vocabulary keyword search.

2.3. Query-by-Example Spoken Term Detection

Query-by-Example (QbE) Spoken Term Detection (STD) is locating a short spoken query inside a larger speech corpus based on template matching techniques. Contrary to keyword search or keyword spotting where the queries are in textual form, the queries in QbE-STD are in the form of spoken utterances. Most applications of QbE-STD rely on (i) feature extraction to represent short temporal windows of speech

in both utterances and (ii) apply a form of Dynamic Time Warping (DTW) algorithm to match a subsequence of the test utterance with the query. Segmental DTW [25], subsequence DTW [26], slope constrained DTW [27] are all examples of dynamic time warping algorithm which are used in subsequence matching and query-by-example audio retrieval. More recently, following the general trend towards deep learning, Convolutional Neural Network (CNN) based matching [28] and end-to-end methods [29] started to become more prominent in the QbE-STD domain.

2.4. Neural Machine Translation

Sequence-to-sequence machine translation has long been the primary goal of researchers to remove the language barrier. Although early approaches to machine translation involve rule based and phrase-based statistical modeling strategies, the main leap is achieved with the help of neural networks in recent years. Neural machine translation usually involves encoding the source sentence in a context vector then decoding into another sequence in the target language. These encoder-decoder networks are usually comprised of a variation of recurrent neural networks (RNNs) such as Long Short Term Memory (LSTM) [30] or Gated Recurrent Units (GRU) [31]. Later on, with the incorporation of attention mechanism into these encoder-decoder structures [32], looking at relevant parts of the source sentence became possible for each word in the translation, which increased the translation performance. More recently, using self-attention to replace recurrent networks for language modeling is also becoming mainstream [33].

An important application of neural machine translation is in the domain of sign language. Since sign language is often not exhaustively annotated with gloss level transcriptions, the only labeling we can usually find is translations into the corresponding written language. Thus, due to this label scarcity, training neural machine translation systems for sign language is easier than gloss-level sign language recognition. Camgoz et al. [34] was the first to attempt neural sign language translation with a bi-directional LSTM model using RGB images to represent sign language video. Later on, Ko et al. [35] used human keypoint estimations in sign language video representation and

Orbay and Akarun [36] experimented with different tokenization strategies. Following the trend in neural machine translation, self-attention has also started to be used in the encoder-decoder structures of sign language translation instead of recurrent neural networks [37].

2.5. Cross-Lingual Information Retrieval

The objective in cross-lingual information retrieval (CLIR) is to search for a query inside a document in a different language. Usually, both the query and the document in CLIR are in the text form. The traditional approach to deal with this problem is to translate either the query or the document, where query translation approach is usually taken due to computational complexities in translating the document [38].Saleh and Pecina [39] use multiple translation hypotheses for the query while searching inside the cross-lingual documents. Vulic and Moens [40] use bi-lingual word embeddings to translate both the query and the document into a common cross-lingual embedding space.

In a more realistic setting, cross-lingual information retrieval is also meaningful when the target language is low resource or the target utterances do not have a transcribed text form. Under such settings, searching for written form queries from a language we know and search in a corpora consisting of unknown languages. The cross lingual keyword search method that enables searching for English queries in a speech corpora consisting of 11 languages by Zhang et al. [41] is an example for this task.

3. QUERY-BY-EXAMPLE SEARCH

Query-by-Example Keyword Search is the search for an example query inside a continuous utterance. In this chapter, we summarize our paper Dynamic Time Warping Based Sign Retrieval [8] that aims for this task. The pipeline can be viewed from Figure 3.1: we first start by extracting frame-wise features from both isolated query video and continuous sign language videos in the corpus. Then, an optional temporal trimming procedure is applied to remove the non-sign beginning and end frames. Finally, isolated samples are searched throughout the corpus with the help of Sub-Sequence Dynamic Time Warping (SS-DTW) algorithm.

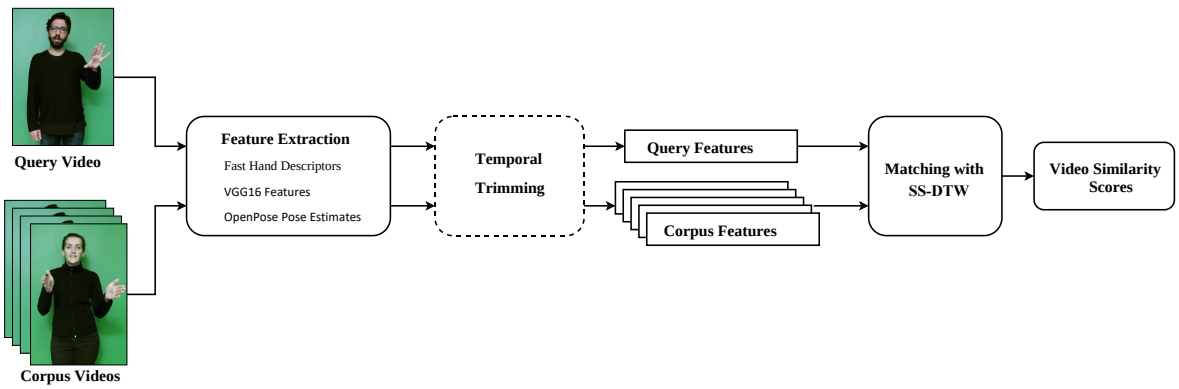

Figure 3.1. Pipeline for our query-by-example keyword search method

3.1. Visual Feature Extraction

With visual feature extraction, the aim is to encode a sign language video into a frame-length $\times$ fixed-size-vector dimensional matrix. Three visual features are selected to represent three different approaches in sign language research. These are: the last layer of VGG-16 convolutional network representing a transfer learning approach, fast hand descriptors representing a pre-deep-learning approach, and openpose pose coordinates representing a deep-learning based skeleton-driven approach.

3.1.1. VGG-16 Convolutional Network

Transfer learning is a well-established technique in the computer vision community which relies on using models developed for certain tasks in another yet similar problem. In this thesis, VGG-16 network [42], a well-known image classification network trained on the ImageNet dataset [43], is used in feature extraction to represent a straightforward transfer learning approach in query-by-example search. 4096-dimensional vectors are extracted from the last layer of 16-dimensional VGG-16 network and a sign language video of N frames is represented with $N \times 4096$ matrix.

3.1.2. Fast Hand Descriptors

Fast hand descriptors are image and video-processing based features introduced for isolated sign language recognition [44]. They are a combination three common feature extraction methods which were widely popular during pre-deep-learning video processing research: Histogram of Oriented Gradients (HOG), Histogram of Optical Flow (HOF) and Motion Boundary Histogram (MBH). For a sign language video comprised of N frames, fast hand descriptors are obtained in the following fashion: Left and right hands are cropped into 80×80 dimensional hand-crops for each frame as the first step. Then, Histogram of Oriented Gradients (HOG) [45] features are extracted for each hand crop, resulting in $N \times 64$ dimensional HOG features. Histogram of Optical Flow (HOF) and Motion Boundary Histogram (MBH) features are obtained from consecutive frame pairs, resulting in $(N-1) \times 72$ dimensional HOF and $(N-1) \times 128$ dimensional MBH descriptors (MBHx and MBHy) for a sign language video of length N. These three features are concatenated by dropping the first frame HOF descriptor to have the same length and the end-result fast hand descriptors are $(N-1) \times 392$ dimensional.

3.1.3. OpenPose Joint Locations

Skeleton joint locations give important cues for determining human body movements. Furthermore, perception studies show that humans can recognize sign language from seeing only the motion of point light displays, i.e. moving points of light attached to skeleton joints [46]. Thus, skeleton joint locations is another meaningful visual feature to be used in sign language query-by-example search.

In this study, part affinity fields based OpenPose [47] framework is used in skeleton pose estimation. Based on state-of-the-art deep learning methods, this framework can predict x and y spatial locations of body, hands, and face joints from a single video frame. For an N frames long sign language video, the participant's upper body, left hand, right hand, and face keypoints extracted with OpenPose framework. The features are then concatenated to represent the video with $N \times 264$ dimensional keypoint matrix. All the (x, y) keypoint locations are further normalized by fixing the neck keypoint at the origin of the coordinate system and all other keypoints' distance to neck is divided by the signer's neck-to-hip length to mitigate the effect of signer variation.

3.2. Temporal Trimming

Isolated and continuous sign language videos in most datasets start and end with hands in the idle position. When applying pattern matching algorithms, these redundant frames can actually hinder the pattern matching algorithms from spotting the actual sign inside the utterance. Douglas-Peucker [48] line segmentation algorithm is employed to trim the idle-to-start and end-to-idle transitional movements. The movement of right and left hand wrist keypoint locations turned into a 2D curve and then this curve is segmented through Douglas-Peucker algorithm. The first and last segments are trimmed and the remaining segments are used as the relevant parts of the sign.

3.3. Subsequence Dynamic Time Warping

When searching for an isolated sign query $Q = (q_1, ..., q_N)$ inside a longer continuous sign language utterance $U = (u_1, ..., u_M)$, we first start by defining the frame-wise cost matrix $\mathbf{C}_{N \times M}$ between sequences Q and U. In this thesis, an element of this matrix $\mathbf{C}(i, j)$ is calculated by using either Euclidean or cosine distance:

$$\mathbf{C}_{euc}(i, j) = \sqrt{q_i^2 + u_j^2} \tag{3.1}$$

$$\mathbf{C}_{cos}(i, j) = 1 - \frac{q_i \cdot u_j}{\|q_i\| \|u_j\|} \tag{3.2}$$

From the cost matrix $\mathbf{C}$, we find the similarity between two sequences of different length using a non-linear alignment generated by sub-sequence dynamic time warping (SS-DTW) [26]. The alignment path $\phi = (\phi_1, ..., \phi_t)$ is defined by ordered pairs along sequences $Q_{1:N}$ and $U_{1:M}$:

$$\phi = \{(n_t, m_t)\} \quad , \quad t = 1, ..., T \quad , \quad n = 1, ..., N \quad , \quad m = 1, ..., M \tag{3.3}$$

and the alignment path is monotonically increasing:

$$\phi_k - \phi_{k-1} \in \{(0, 1), (1, 0), (1, 1)\}$$

In SS-DTW, we search for the entire length of the isolated sign sequence $Q = (q_1, ..., q_N)$. Thus, the warping path should encompass the entire length of the short sequence, i.e. $\phi_1 = (1, i \geq 1)$ and $\phi_T = (N, j \leq M)$. With these two constraints, we find the alignment path ϕ and the cumulative distance matrix $\mathbf{D}$ from the cost matrix $\mathbf{C}$ using the algorithm described in Figure 3.2. Then starting from $D(N, j)$, optimum path ϕ^* is found by back-tracking with t^*.

SS-DTW$(\mathrm{q}_{1:N}, u_{1:M})$

▷ Initialization

for $m = 1$ to M **do**

$\mathbf{D}(1, m) \Leftarrow \mathbf{C}(1, m)$ ▷ Initialize the first row of cumulative distance matrix to the cost matrix. This makes starting at any position of the utterance possible without accumulating any cost.

end for

for $n = 1$ to N **do**

$\mathbf{D}(n, 1) \Leftarrow \sum_{k=1}^{n} \mathbf{C}(k, 1)$ ▷ Initialize the first column as in traditional DTW

end for

▷ Forward Pass

for $n = 2$ to N, $m = 2$ to M **do**

$\mathcal{T} \Leftarrow \{(n-1, m-1), (n, m-1), (n-1, m)\}$

$t^* \Leftarrow \mathrm{argmin}_{t \in \mathcal{T}} (\mathbf{D}(t))$

$\mathbf{D}(n, m) \Leftarrow \mathbf{D}(t^*) + \mathbf{C}(n, m)$

end for

Figure 3.2. Sub-sequence dynamic time warping (SS-DTW) algorithm.

4. KEYWORD SEARCH USING MACHINE TRANSLATION

Unlike spoken languages where one can easily find a direct transcription of the speech utterance in a written form, sign language data generally do not have such aligned transcriptions. However, the data is abundant in the form of sign language interpretations accompanying news, movies, shows, and TV series that are originally in the spoken language. In this chapter, we summarize our machine translation based cross-lingual keyword search technique that enables retrieving sign language utterances with translated queries [9]. With the help of machine translation, this technique is capable of learning separate language models for the signed and the spoken languages, which differ in their vocabulary and grammar.

Our model follows a two stage approach similar to Lattice-based Search for Spoken Utterance Retrieval [20], but differs in that we use beams generated from the decoder of a neural machine translation network rather than HMM lattices. At the first stage, we use the encoder-decoder network from Camgoz et al. [34] to translate the sign language videos into written language word sequences. In doing so, different possible translations (beams) are stored with corresponding probabilities rather than simply picking the translation with the highest probability. Then, at the second stage, probability of a word being in the video is obtained through finding the expected count of the word in all the stored translation beams. Implementation details can be found in this chapter. For cross-lingual keyword search results obtained with this model, please visit Section 6.5.

4.1. Neural Machine Translation Model

We implemented the sign-to-text (S2T) translation network from [34] that directly translates the frames of the sign language video into written language. In this section, we give a brief summary of this S2T neural translation model and explain how we

obtain decoder beams with beam probabilities. The model is comprised of three parts: 1) video and word embeddings, 2) encoder-decoder network, and 3) beam search.

4.1.1. Video and Word Embeddings

All neural machine translation (NMT) models start with embedding the input and output sequences into a space where the encoder and decoder can operate. In this work, the target sequence is modeled as a written language with words in the vocabulary as the units. For this purpose, word embeddings [49] are used for the target language as standardized in the literature. The source sequence, however, is from a visual language and using word embeddings is not possible. Because of this, video frames are embedded into vectors through AlexNet [50] as the tokenization layer. The weights of the AlexNet model are initialized by pretraining on the ImageNet [43] database and then finetuned together with the encoder-decoder network.

4.1.2. Encoder-Decoder Network

The encoder-decoder modules of [34] are both based on Long Short Term Memory (LSTM) [51] with an attention layer [52] between the encoder and the decoder. With this structure, sign language videos of different frame lengths can be converted into translations of different word counts. Moreover, thanks to attention, focusing at different video parts is possible when generating different words. For detailed explanation, please visit neural sign language translation by Camgoz et al. [34]. In this thesis, their S2T network with their best performing hyper parameters are trained and used for test time decoding. Thus, only the parts relevant to the decoding are discussed here.

With $x_{i\in 1:T}$ denoting the embedded source (sign language video) sequence and $y_{j\in 1:N}$ denoting the embedded target (translation into the written language) sequence, the encoder-decoder structure is trained to maximize the conditional probability of the video producing the correct target sequence $p(y|x)$. Since the target is a causal sequence, this conditional probability can be opened as

$$p(y|x) = \prod_{n=1}^{N} p(y_n|y_{1:n-1}, x_{1:T}) \tag{4.1}$$

$$= p(y_N|y_{1:N-1}, x_{1:T}) \prod_{n=1}^{N-1} p(y_n|y_{1:n-1}, x_{1:T}) \tag{4.2}$$

and can be trained to minimize the cross-entropy between each word the neural machine translation model produces (y_n) and the matching correct word in the target sequence.

In test time, each word in the output sequence y_n is obtained using the entire input sequence $x_{1:T}$ and the predicted output sequence up to that time $y_{1:n-1}$. From Equation 4.1, we see that the predicted sequence maximizing the probability $p(y_{1:n-1}|x_{1:T})$ does not necessarily maximize $p(y_{1:n}|x_{1:T})$ after a new y_n element is considered. Thus, the predicted sequence maximizing $p(y|x)$ can change after every time step.

4.1.3. Beam Search

The main motivation behind using beam search in the decoder is to store k predictions with highest probability rather than storing only the single best prediction at the current time step y_n. By doing so, we increase the chance of having the sequence maximizing $p(y_{1:n+1}|x_{1:T})$ already stored. Thus, at every time step, we have a beam of k best predictions stored and update those k beams at each step consecutively. For a good translation model that can maximize the conditional probability $p(y|x)$, it is expected that with increasing beam size k, the chance of not eliminating the maximizing sequence also increases.

Beam search is generally used to carry the best translation sequence to the end. After the final step, the sequence with highest probability is chosen as the prediction and the remaining $k-1$ beams have no further use. In this work, we use all the k beams $y^1, ..., y^k$ with their respective conditional probabilities $p(y^1|x), ..., p(y^k|x)$ for calculating the expected count of keywords across all the k possible translations.

4.2. Keyword Search from Decoder Beams

A common approach in keyword search for spoken utterance retrieval is searching for the keyword in a lattice generated by Automatic Speech Recognition (ASR) systems [20] and indexing the keywords with expected counts along lattices. In this thesis, a similar method is applied to neural machine translation outputs. The probability of a keyword being in a sign language utterance (video) is found by calculating the expected count of that word along the decoder beams of the translation model. Then, retrieval performance at different operating points is calculated by comparing these probabilities against threshold values in the range $0 - 1$.

4.2.1. Calculating Expected Counts

With r denoting a possible translation for the sign language video and w a word in this translation, the count of a keyword q in this sequence $C(q|r)$ is found by counting how many times the keyword q is repeated:

$$C(q|r) = \sum_{w \in r} \mathbb{1}(w = q) \tag{4.3}$$

For a decoder beam B comprised of k different possible translations, the expected count of the keyword q is calculated by taking a weighted average of keyword counts $C(q|r)$ in different translations, with the probability of that translation being correct $p(r)$ as weights:

$$C(q|B) = \sum_{r \in B} p(r) C(q|r) \tag{4.4}$$

$$= \sum_{r \in B} p(r) \sum_{w \in r} \mathbb{1}(w = q) \tag{4.5}$$

4.2.2. Beam Normalization

In beam search, only k-best translations with the highest probability are stored and the remaining paths are eliminated. Thus, sum of probabilities for the stored sequences are smaller than one $\sum_{r=r^1}^{r^k} p(r) < 1$. This can create some problems if the generated sequence lengths are too big and the stored subset is small compared to all possible paths. To remedy this, the probabilities of stored paths $p(r)$ are normalized so that they sum up to 1:

$$p(r^{'}) = \frac{p(r)}{\sum_{\tilde{r} \in B} p(\tilde{r})} \tag{4.6}$$

Since the beam width k and beam normalization can change the success of retrieval, the experimental results reported in Section 6.5 are with different beam widths and normalization options.

5. KEYWORD SEARCH WITH NEURAL EMBEDDINGS

With the end-to-end structures being ever more popular in the realm of computer vision and natural language processing, the same approach also gained a foothold in spoken utterance retrieval. Audkhasi et al. [53] were the first to remove Automatic Speech Recognition from the keyword search procedure. Later on, Shan et al. [24] used attention for the similar task of end-to-end keyword spotting. Following the trend in spoken utterance retrieval, we introduced an end-to-end keyword search method for sign language based on attention and neural embeddings [10]. Although we do not use raw video frames as the input, this method can be considered end-to-end as much as the aforementioned spoken utterance retrieval techniques since everything is trained together except framewise feature extraction from sign language videos. In this chapter, we give the structural details of this model.

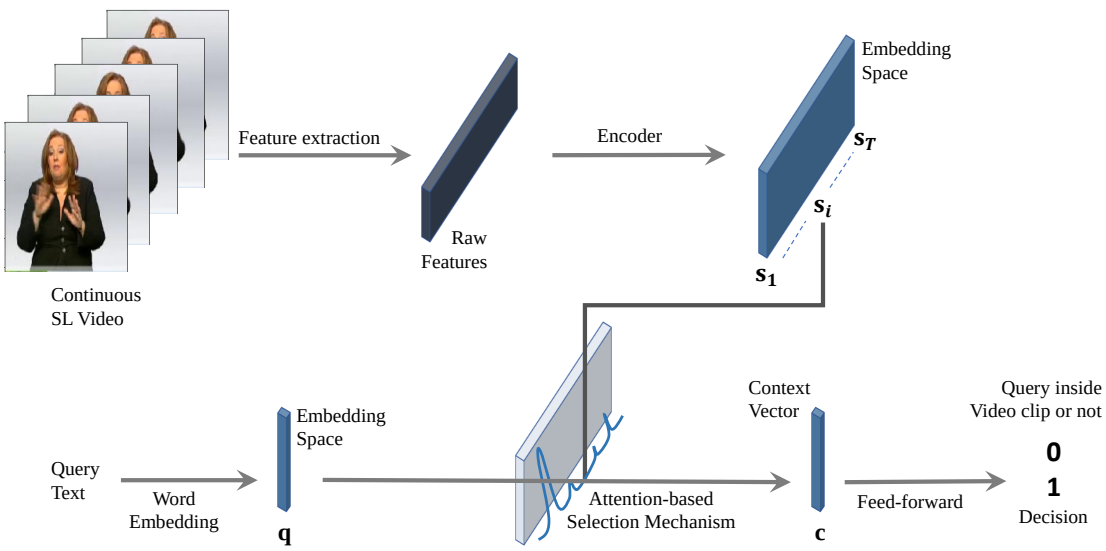

Figure 5.1. General pipeline for keyword search with neural embeddings

The pipeline in Figure 5.1 summarizes our general methodology that can work with different preprocessing techniques and sequence encoders. Firstly, the frames of continuous sign language video are passed through a feature extraction procedure. After raw features are obtained from each frame, the rest of the structure is trained end-

to-end to predict whether the keywords are present in the video or not. Raw features and the queries are embedded into the same embedding space and query vectors q are searched inside the sequence $s_{1:T}$ with the help of attention based selection mechanism. Then, from the obtained context vectors c, the keyword search decision is made.

We used Binary Cross-Entropy (BCE) as our target function during training. One important problem in training of this keyword search model was to ensure a stable convergence. In our initial models where we did not search for all the queries in our vocabulary simultaneously, the training did not converge to a stable point. We remedied this by searching for all the queries simultaneously regardless of how many of them are positively tagged for the sign language video. For example, when we have 1000 keywords in our vocabulary and the current sign language video contains 8 of them, our target in train phase is to match with 8 positive and 992 negative labels. This training procedure ensures parameter sharing for sequence encoder whilst word embeddings for each query can be learned independently. While there is a single $s_{1:T}$ matrix, for each of the 1000 keywords $q_1, q_2, ..., q_{1000}$; 1000 different context vectors $c_1, c_2, ..., c_{1000}$ are generated. With Binary Cross-Entropy as the target function, 8 of these context vectors are trained to match the target "1", and the other 992 context vectors are trained to match the target "0".

The rest of this Chapter is constructed as follows. In the first three sections, the building blocks of the aforementioned model are explained in detail: the feature extraction techniques used are discussed in Section 5.1; different sequence encoders matching with these features are discussed in Section 5.2; and the selection mechanism introduced in Section 5.3. Using these building blocks, a brief overview of our Handshape-based and Pose-based Keyword search structures can be found in Section 5.4. In the last two sections of this chapter, the techniques we used to improve keyword search performance with these models is given. In Section 5.5, a cold fusion strategy for combining different keyword search models is introduced. Finally, in Section 5.6, we share the adaptations we used to increase the cross-lingual keyword search performance.

5.1. Feature Extraction

5.1.1. Background

Being a visual language, the information in sign language is mainly carried through a mixed use of hand movements and facial expressions. To cope with this multimodal nature, recognition studies in sign language have historically divided the problem and focused on these different building blocks separately. Hand shape recognition from RGB hand patches [36,54,55] models common hand shapes, body pose based sign language recognition [56,57] mainly models places of articulation in space or along body, and other studies deal with mouthing [58] and facial expressions [59] which are all important communication channels of continuous sign language. In this part of thesis, we focused on the two main components which are generally believed to convey much of the information in sign language: hand shapes and places of articulation. We used DeepHand and Multitask convolutional neural networks (CNNs) for extracting hand shape features and OpenPose pose estimation framework for extracting information about places of articulation in space or along body.

5.1.2. Pose Keypoint Extraction using OpenPose

Part affinity fields based OpenPose framework [47] is used to extract 2D pose estimates of upper body, right and left hand. Upper body pose estimates are obtained from 25-keypoint body model of the OpenPose framework by selecting the 13 keypoints that are above the hipline. For right and left hand joints, 21-keypoint hand model from OpenPose is used. An example sequence with OpenPose pose estimates projected on top can be viewed in Figure 5.2.

In OpenPose framework, the (x, y) coordinate estimates are provided with the model's confidence on finding the keypoints. In blurry and low resolution datasets, especially, the pose estimation process cannot always result in good coordinate estimates for each joint and confidence scores provide a meaningful information regarding how

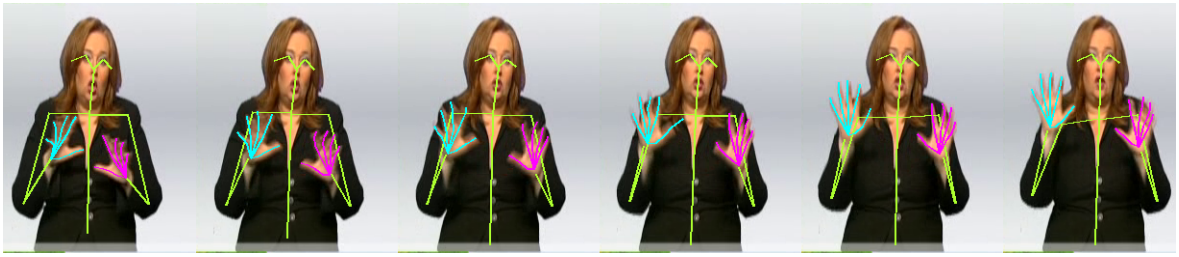

Figure 5.2. An example sign language sequence with OpenPose upper body, right hand, and left hand skeletons shown on top. The extracted skeletons are colored with yellow for upper body, cyan for right hand, and magenta for left hand, respectively.

much we should rely on a specific keypoint estimate. Due to this reason, we also used the related confidence scores along with (x, y) spatial locations in our experiments.

5.1.3. Hand Shape Feature Extraction

Our hand shape feature extraction method can be seen in Figure 5.3. For each frame, the right hand wrist spatial locations are extracted with the help of OpenPose [47] pose estimation toolkit and square hand crops centered around the right hand wrist joint are obtained. By feeding hand crops into one of the two pre-trained 2D CNN options, we represent hand crop with a vectoral feature. The frames are omitted if the OpenPose cannot estimate the location of the wrist joint. For the two 2D CNN options, the resulting one-dimensional hand shape features are 1024-dimensional for DeepHand and 2048-dimensional for MultiTask respectively.

5.1.3.1. Hand Shape Feature Extraction using DeepHand CNN. We used the pre-trained CNN from DeepHand [54] to extract hand shape features. The model takes hand crops and classifies them into 60 pre-defined hand shape classes or a junk class. Their training data consists of two isolated sign language corpora (Danish and New Zealand SL), and continuous Phoenix-2014 Weather dataset. Since the third dataset they used in training is almost identical to our experiment data and the amount of supervision they used in training is more than that of our keyword search models, we believe the pre-trained DeepHand model can be viewed as the topline for hand shape encoders in this dataset.

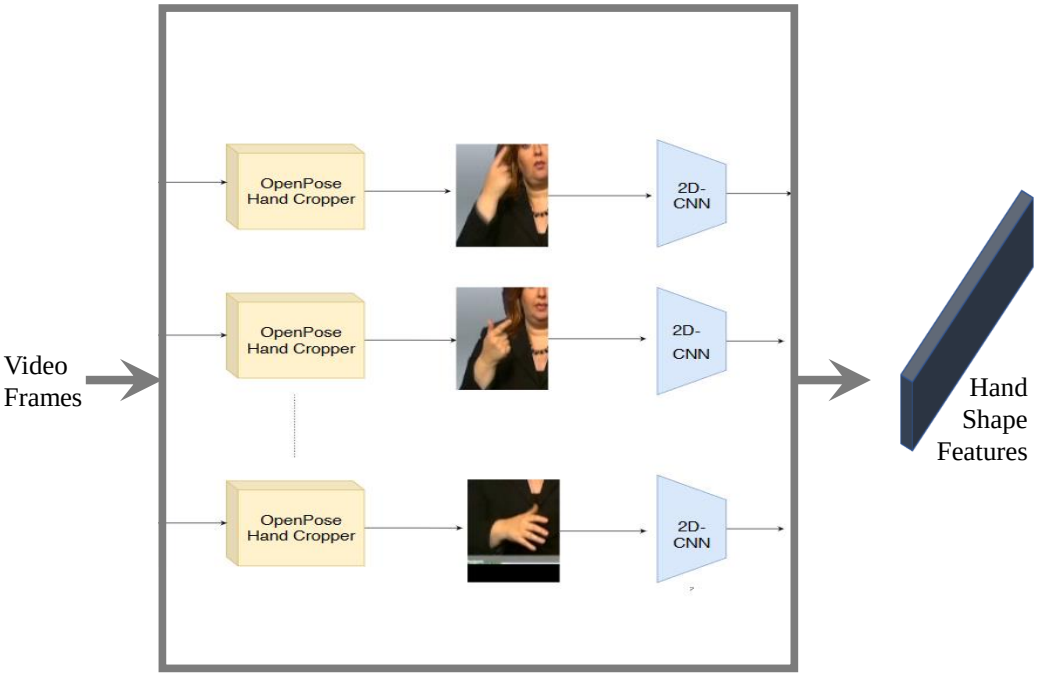

Figure 5.3. Hand shape feature extraction procedure. After the right hand crops are obtained, they are fed into either DeepHand or Multitask CNNs.

In our implementation, we used 1024 dimensional features from the second-last layer of DeepHand CNN.

5.1.3.2. Hand Shape Feature Extraction using Multitask CNN. Multitask features are introduced as a tokenization layer for sign language translation [36]. The network is trained for hand shape recognition in two datasets: the first one is the Danish and New Zealand SL corpora from DeepHand [54] excluding RWTH-PHOENIX-Weather 2014, and the second one is a framewise labeled and smaller Turkish SL dataset [60]. The network shares parameters at the start, and the final layers are different for matching different hand shape classification tasks. While the first one is 60 hand shapes and a junk class, the target for the smaller dataset also includes specific classes for hands showing certain body parts, thus, incorporating background information to some extent. Since the domain data is not used in training of Multitask features, it can be thought of as a real-world scenario for RGB hand shape based KWS. In feature extraction, we used 2048 dimensional vectors from the shared part of the multitask network.

5.2. Encoder Structures

In our experiments, we initially started with Recurrent Neural Networks (RNNs) for the sequence encoders but those models were not successful. We later on found that encoder structures based on some sort of temporal convolution perform better when used with our selection mechanism. In this section, we introduce our two sequence encoders that are both based on Convolutional Neural Networks (CNNs): When our input features are skeleton sequences, we use Spatio-Temporal Graph Convolution encoders [61]. When our input features are vectoral hand shape features, we use 1D Temporal Convolutional Networks. These modules are trained together with the keyword search module from Section 5.3 and without any pre-training. The structural details regarding these two modules can be found in this section.

5.2.1. Spatio-Temporal Graph Convolutional Network (ST-GCN) Encoder

When we have the skeleton pose estimates as the input features, we use an encoder based on Spatial Temporal Graph Convolutional Networks (ST-GCN) [61], which is first introduced for the skeleton-based action recognition. In ST-GCN, sequences of 2D or 3D body joints representing the video are first converted into spatio-temporal graphs with the keypoints as the graph nodes and bones + temporal connections as the graph edges (see the connected graph in Figure 5.4). Then 12 layers of ST-GCN operations take place to convert the graph into a vectoral time sequence.

5.2.1.1. Graph Construction. In the original implementation of ST-GCNs [61], full-body OpenPose model without hands are taken as the input to form spatio-temporal graphs. With the keypoints as graph nodes, two types of connections are represented as the edges: 1) spatial graph edges aimed at modeling the bones connecting two joints, and 2) temporal graph edges that connect the same joints through time. In order to use this structure in the task of sign language recognition, we first cropped the keypoints below the hip since they convey no information in sign language. Furthermore, we incorporated hands into the layout through connecting arm joints to the hand wrist

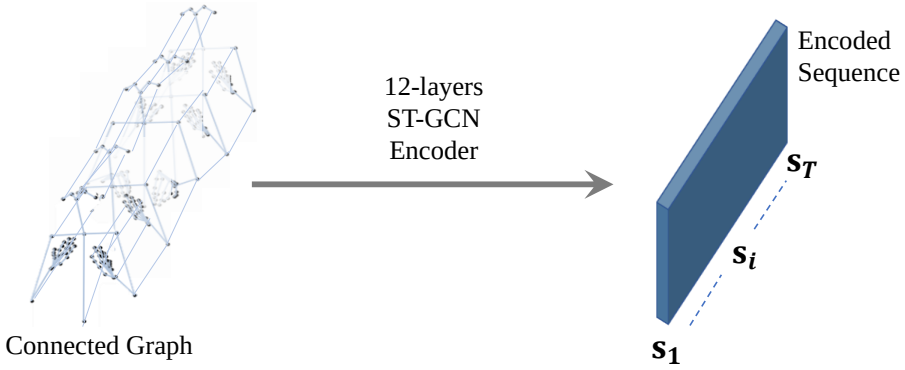

Figure 5.4. The input to the ST-GCN encoder is a connected graph representing the entire sign language sentence. The encoder then converts this graph into a 2D-matrix.

joints; hence the whole layout remains a single graph. The spatial connections of three layout settings we experimented with can be seen in Figure 5.5. Each graph node in the figure is represented with either (x, y) spatial locations, or (x, y, c) including OpenPose estimation confidence c as the third dimension.

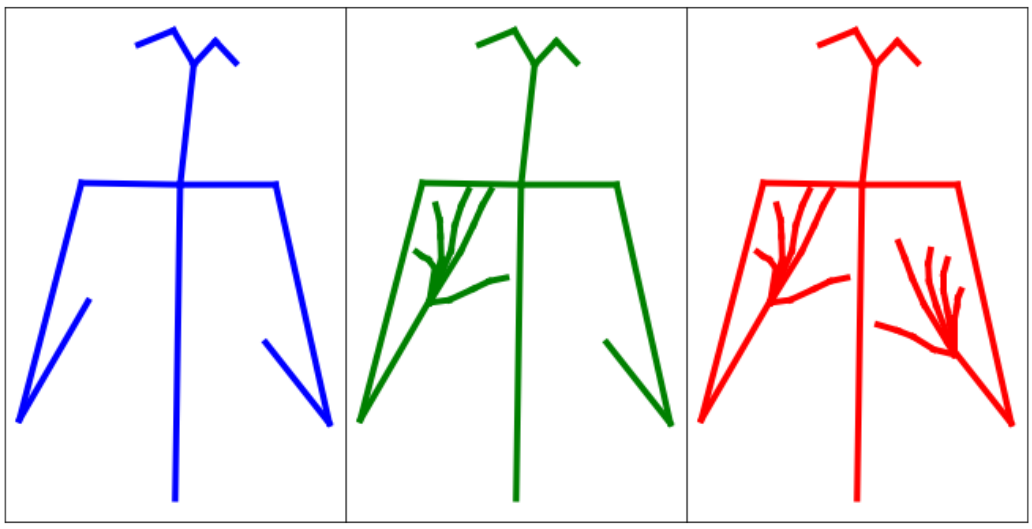

Figure 5.5. The three graph layout options with spatial connections: upper body (13 keypoints), upper body with right hand (34 kp), and upper body with both hands (55 kp). Temporal connections omitted for illustration purposes.

5.2.1.2. One layer of ST-GCN. Our Spatio-temporal graph convolutional network encoder for sign language keyword search is comprised of 12 identical layers of ST-GCN units. In this part, the mathematical background of a single ST-GCN layer is given.

With skeleton joints H_{in}, the graph convolution operation on an undirected graph is approximated as [62]:

$$H_{out} = \sigma(\tilde{D}^{\frac{1}{2}}\tilde{A}\tilde{D}^{\frac{1}{2}}H_{in}W) \tag{5.1}$$

where $\tilde{A}$ is the adjacency matrix , $\tilde{D} = \sum_j \tilde{A}_{ij}$ is the diagonal matrix, σ is the ReLU activation function, and W is the trainable weight matrix of the 2D convolution.

The adjacency matrix $\tilde{A}$ can be formed most straightforwardly as $\tilde{A} = A + I_N$, by summing identity matrix I_N and single frame bone connections matrix A. We further utilized two methodologies described in [61] in forming adjacency matrix: 1) spatial configuration partitioning, which is changing the weights depending on the position of the respective node to the root node, and 2) altering the number of adjacent nodes by adding the second/third order neighbors.

5.2.1.3. Encoder structure. The encoder is comprised of 12-layer stacked ST-GCN units. The first ST-GCN takes the three-channel (x,y,conf) skeleton sequence as input. Then, the number of output channels are 64 for the first five layers, 128 for next four layers, and 256 for the final three layers. As a final step to the encoder, mean pooling over graph nodes is applied so that the encoder result will be [sequence length x 256].

The main difference of our implementation compared to original ST-GCN [61] is the reduction in temporal kernel size. Compared to action recognition where the entire video is comprised of the same action, a sign corresponding to a keyword is rather focused on a small temporal window of the sign language sentence. With each layer having a temporal kernel size of three and no temporal pooling, the total receptive field at the end of encoder becomes 25 frames.

5.2.2. 1D Temporal CNN Encoder

When we have either DeepHand or Multitask hand shape features as the input, we use 1D Temporal CNN Encoder at the sequence encoder. At the end of hand shape feature extraction step step, we obtain each frame represented with raw hand shape features. Since duration of a sign is greater than a single frame, however, we cannot learn keyword embeddings with these raw hand shape features and a further sequential modeling step is necessary. To model a one-second-long temporal sliding window, we used four-layer 1D convolutional network with kernel size seven and same padding. We used leaky ReLU as the activation between layers. The first layer has 1024 channels for DeepHand and 2048 for MultiTask features. Then the channel sizes at the end of each layer are 512 for layer one, 256 for layer two, 128 for layer three and 256 for the last layer, respectively. With the help of same padding during convolution, we kept the encoded sequence length the same with raw hand features. Each time step at the encoded sequence has access to 25 time steps of raw hand shape features resulting in a temporal range of one second in 25-fps RWTH-PHOENIX-Weather 2014T dataset.

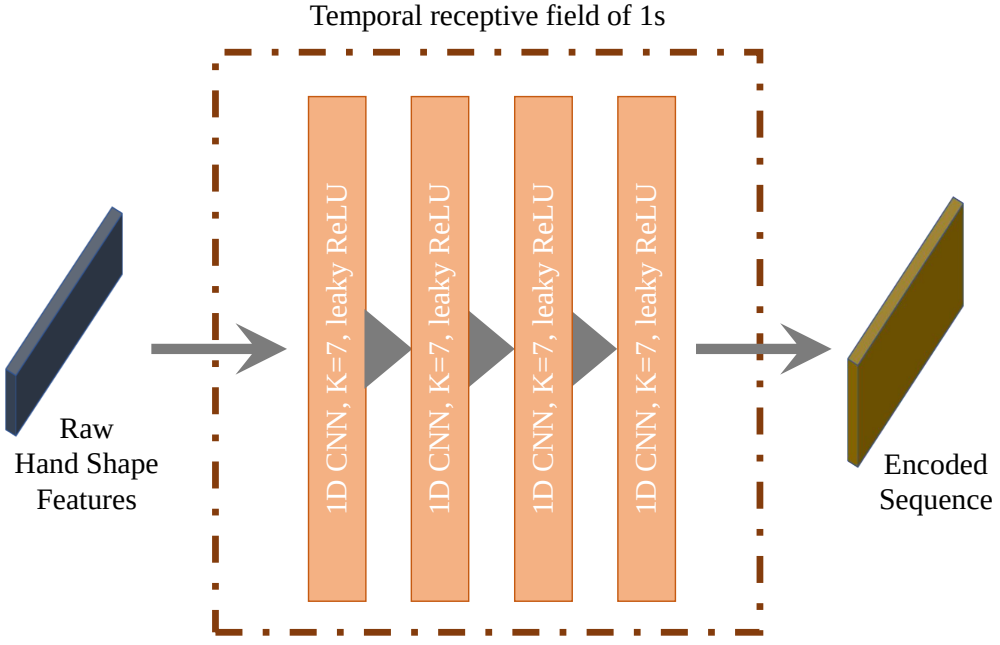

Figure 5.6. 1D Temporal CNN Encoder that we use with hand shape features

5.3. Keyword Search Module

The keyword search module in our implementation can be seen analogous to a decoder in a neural machine translation approach. With the help of keyword search module, we can map both the keywords in the text form and features coming from the sign language sequence into the same embedding space and train word embeddings together with the sequence encoder. The structure of keyword selection mechanism is the same in both hand shape based and pose based keyword search. The keyword search module and consists of word embeddings, attention-based selection mechanism, and the final feed-forward layer.

5.3.1. Word Embeddings

A query in the form of text is first converted into an index in the vocabulary, and for all unique queries, a simple 256-dimensional linear word embedding $\mathbf{q}$ is learned to match encoded sequence frame $\mathbf{s_i}$ in a mutual embedding space. In our implementation, no pretrained word embeddings are used and it is not possible to search for out-of-vocabulary keywords since we only learn word embeddings for in-vocabulary keywords.

5.3.2. Attention-based Selection Mechanism

Attention is a widely known concept from Neural Machine Translation, where it helps the decoder to focus on relevant parts of the source sentence when predicting the next word in the target sequence [52]. In a similar fashion, we use attention to focus on the most relevant part of the encoded sequence $\mathbf{s_{1:T}}$ to the query $\mathbf{q}$. The relevance $score(\mathbf{q}, \mathbf{s}_i)$ between the ith element of the encoded sequence $\mathbf{s_i}$ and the query $\mathbf{q}$ is measured with three different scoring functions: dot product in Equation 5.2, cosine similarity in Equation 5.3, and cos^2 with learnable weights in Equation 5.4.

$$score(\mathbf{q}, \mathbf{s}_i) = \mathbf{q} \cdot \mathbf{s}_i \tag{5.2}$$

$$score(\mathbf{q}, \mathbf{s}_i) = \frac{\mathbf{q} \cdot \mathbf{s}_i}{\|\mathbf{q}\| \cdot \|\mathbf{s}_i\|} \tag{5.3}$$

$$score(\mathbf{q}, \mathbf{s}_i) = \beta \left[\frac{\mathbf{q} \cdot \mathbf{s}_i}{\|\mathbf{q}\| \cdot \|\mathbf{s}_i\|} \right]^2 + \theta \tag{5.4}$$

Using a $score(\mathbf{q}, \mathbf{s}_i)$ function, the context vector $\mathbf{c}$ is obtained from the weighted average of relevance scores after $softmax$ layer:

$$\mathbf{c} = \sum_i \left[\frac{\exp\left(score(\mathbf{q}, \mathbf{s}_i)\right)}{\sum_{i'} \exp(score(\mathbf{q}, \mathbf{s}_{i'}))} \right] \cdot \mathbf{s}_i \tag{5.5}$$

Once the context vector $\mathbf{c}$ is obtained, it is then fed into a one-layer feed-forward network with sigmoid activation to decide whether the query $\mathbf{q}$ is found inside the weakly-labeled sequence $\mathbf{s}_{1:T}$.

Although it only has β, θ, and the weights of the final feed-forward layer as the learnable parameters, the selection mechanism is the most important layer of the network since it makes the weakly supervised learning of keyword embeddings possible. All the keywords in the vocabulary are searched in tandem in the same sequence. The keywords that appear in the transcription sequence are labeled positive whilst keywords that are not apparent in the transcription are trained to match negative labels.

5.4. Summary of Pose Based and Hand Shape Based Keyword Search

With the building blocks described in Sections 5.1-5.3, we can construct pose based and hand shape based KWS models. In this section, we give a brief overview of both models. The pipeline of our pose based KWS system is summarized in Figure 5.7. Our method starts with extraction of skeleton keypoints for upper body, right hand, and left hand, as described in Section 5.1.2. Choosing one of the layout options from Figure 5.5, we form connected graphs, and encode the video into the embedding space with Spatio-temporal Graph Convolution Encoder from Section 5.2.1. Keyword selection module from Section 5.3 represents the keywords in the same embedding space and focuses on relevant parts of the encoded hand shape sequence to detect the keyword. We search for all the keywords in our vocabulary at the same time. All the modules seen in the pipeline are trained end-to-end except pose estimation.

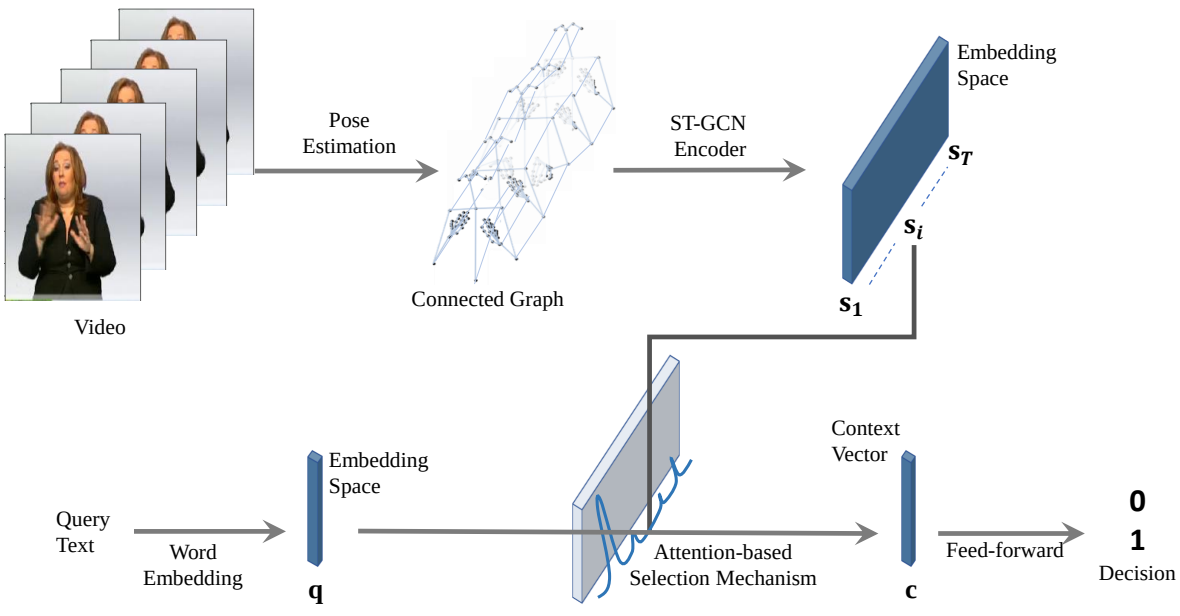

Figure 5.7. Pipeline for pose estimation based keyword search with neural embeddings

The pipeline of our hand shape based KWS system is summarized in Figure 5.8. This method starts with feature extraction from the video to obtain hand shape feature vectors for each frame, as described in Section 5.1.3. In doing so, either DeepHand

features or Multitask features are used. Frame-level hand features are then fed into a four-layer 1D temporal CNN encoder from Section 5.2.2 to detect movements. Keyword selection module from Section 5.3 represents the keywords in the same embedding space and focuses on relevant parts of the encoded hand shape sequence to detect the keyword. We search for all the keywords in our vocabulary at the same time. All the modules seen in the pipeline are trained end-to-end except hand shape feature extraction.

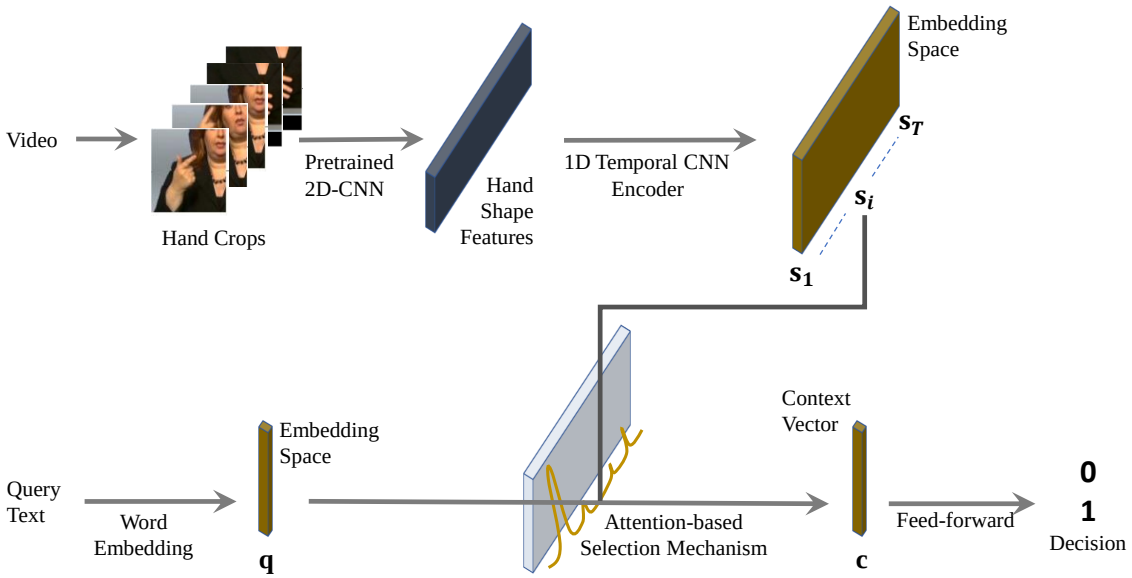

Figure 5.8. Pipeline for hand shape based keyword search with neural embeddings. The main difference of this method from the one in Figure 5.7 is that instead of ST-GCN, a simple 1D Temporal CNN is used in the encoder.

5.5. Fusion Strategy

Hand shape and pose keypoints are among the most common pretrained features used in sign language recognition studies. However, a KWS model trained with an encoder based on whole skeletal graph focuses on very different specifications compared to a KWS model based on vectoral features obtained from only right hand. Thus, it is possible to obtain a higher-performing ensemble model by fusing the two decisions. In this thesis, a late fusion approach is applied to test the effectiveness of combining keyword search models trained with hand shapes and pose keypoints.

In a setting where we search for a single keyword in a single sign language utterance, let $l \in 0, 1$ represent the binary label, h the prediction of the hand shape based KWS model, and k the prediction of the pose keypoints based KWS model respectively. The fused prediction p is found as

$$\log p = (1 - \gamma) \cdot \log h + \gamma \cdot \log k \tag{5.6}$$

The blending ratio γ is the number maximizing the retrieval metric (mean average precision-mAP score for this study) in train and development sets and fusion results are reported using this γ-value in the test set.

5.6. Cross-Lingual Keyword Search with Neural Embeddings

Using the framework described in Sections 5.1-5.4, it is possible to retrieve sign language sentences with queries in the form of either glosses or cross-lingual keywords. However, as we already discussed in Chapter 4, sign language has different vocabulary and grammar than the written language, and using cross-lingual written queries without any language modeling results in sub-par performance. Words in the written language do not match one to one with their sign language glosses. Furthermore, for some less frequent words, the predictions we get by training with only small amount of data are most often unreliable. To remedy these problems, we introduced two rescoring based context modeling strategies for sign language in our paper Cross-Lingual Keyword Search for Sign Language [11].

Rescoring keyword search predictions with the predictions for other keywords [63] and the predictions of the same keyword at another close time instant [64] is a known strategy in spoken keyword search. In this work, we define two prediction rescoring strategies that use the model's own predictions for other queries within the vocabulary V. For a single sign language sentence, let $\vec{l}$ represent the $|V|$ dimensional correct labels, s.t. we have one label $\in \{0, 1\}$ for each query, and $\vec{p}$ represent the $|V|$ dimensional

vector comprised of the trained model's predictions. Our aim is to come up with a new predictions vector $\vec{\tilde{p}}$ that is better than the original $\vec{p}$. We do this by two different strategies: (i) a statistical context model based on bag-of-words TF-IDF vectors, and (ii) a machine-learning based multi-layer perceptron (MLP) context model.

5.6.1. Statistical Context Modeling with TF-IDF Vectorization

Term Frequency Inverse Document Frequency [65] vectorization is a well known strategy for language modeling. While calculating a weight for a query inside a document, this algorithm gives high weights to queries that are seen multiple times in this document (high term frequency), and low weights to ones that are seen in many other documents (inverse document frequency). Thus, by finding the similarity between each keyword in the vocabulary V and the document, we obtain a $|V|$ dimensional vectoral representation of the document. Let $\vec{d_i}$ be the l_1 normalized TF-IDF vector for the ith document in our training set, the document context model $\vec{d_q}$ for query q is found by averaging over all the documents in our training set that contain this specific query:

$$\vec{d_q} = \text{avg}(\vec{d_i} : \text{tfidf}(q, \vec{d_i}) > 0) \tag{5.7}$$

Then, by looking at the cosine similarity between this query-specific document context vectors $\vec{d_q}$ and our model's prediction vector $\vec{p}$, we obtain the new scores. The new prediction score for the ith query in the vocabulary $\tilde{p}(q_i)$ is formulated as

$$\tilde{p}(q_i) = 1 - \frac{\vec{d_q} \cdot \vec{p}}{|\vec{d_q}||\vec{p}|} \tag{5.8}$$

and the new prediction vector $\vec{\tilde{p}}$ is simply the new prediction values for all the queries in our vocabulary V.

$$\vec{\tilde{p}} = \left[\tilde{p}(q_1), ..., \tilde{p}(q_i), ..., \tilde{p}(q_{|V|})\right]^\intercal \tag{5.9}$$

The statistical context modeling for the query, by itself, does not give better results than the model's own predictions. However, when combined with the original predictions through a hyperparameter, it boosts the prediction scores. We applied the same cold fusion strategy from Section 5.5 in combining pre-context modeling predictions and predictions obtained with statistical context model.

5.6.2. Multilayer Perceptron Based Context Model

For a sign language sentence in our training set, we trained a simple multi-layer perceptron with $|V|$ dimensional predictions vector $\vec{p}$ as the inputs and labels vector $\vec{l}$ as the target. The network is comprised of two hidden layers of size 256 with ReLU activations and a dropout probabilty of 20%. We finished the training with early stopping when the loss in the development set was not decreasing any further.

6. EXPERIMENTS AND RESULTS

6.1. Datasets

Three datasets are used throughout this thesis. The first one, HospiSign [66], is a dataset in Turkish Sign Language, and we used it in measuring the success of our Query-by-Example Keyword Search (KWS) model from Chapter 3. The second one, RWTH-Phoenix-Weather 2014T [34], is a dataset in German Sign Language, and it is the main dataset we experimented with in both KWS using Machine Translation from Chapter 4, and KWS with Neural Embeddings from Chapter 5. Lastly, MeineDGS dataset [67] is also in German Sign Language, and is used to measure KWS with Neural Embeddings models from Chapter 5. In all three datasets, we train/test using weak labels, i.e. no gloss level temporal annotations are utilized.

6.1.1. HospiSign

For Query-by-Example Keyword Search, we used BosphorusSign [68], a Turkish Sign Language recognition dataset which includes RGB sign language videos, their depth maps, and skeleton pose information alongside translations into Turkish language. In BosphorusSign dataset, there are isolated and continuous sign language utterances, each repeated by 8 different signers. The dataset is divided into four groups: three of these contain isolated sign language classes in the domains of finance, health, and frequently used signs. The remaining part, named HospiSign [66], is collection of continuous sign language sentences comprised of typical conversations between medical personnel and the hearing impaired. In this work, we used 57 isolated sign classes from health section of BosphorusSign dataset together with 41 continuous sign language sentence classes of the HospiSign dataset.

Since there is no query-by-example keyword search procedure defined on this data, we defined our own procedure. Samples from 57 isolated signs are used as citation-form

queries and searched inside 41 continuous sign language sentences from HospiSign. Each continuous sign language sentence is labeled with 1 or 0 for each query based on whether the query is inside the Turkish translation of continuous sign language sentence.

6.1.2. RWTH-Phoenix-Weather 2014T

RWTH-PHOENIX-Weather 2014T [34] is the main dataset we used for our experiments. This dataset is originally introduced for translation task and includes weather forecasts in German, their sign language interpreting in video format, and gloss sequence corresponding to the signs in the interpreting. The video footage is in 25 fps and in low resolution with heavy amount of blur. There is 9.2 hours of training, 37 minutes of development and 43 minutes of test partitions in the dataset.

For both the gloss and the cross-lingual keyword search, we used the original dataset labels in the following fashion: We formed our vocabulary from the training set. Each video is weakly labeled with 0 or 1 for every keyword in our vocabulary by looking at whether the keyword is in the label sequence or not. We dropped glosses starting with "__" since they contain on/off tokens and ambiguous signs, and we did not utilize lemmatization for German keywords. At the end, we have 1085 glosses in our gloss vocabulary and 2887 German keywords in cross lingual vocabulary. Since 392 of the glosses and 942 of the German keywords are shared between train and test sets, we report our results on this shared vocabulary. Out-of-vocabulary keyword search is not supported in this implementation. We report our results in this dataset using both KWS using Machine Translation from Chapter 4, and KWS with Neural Embeddings from Chapter 5.

6.1.3. MeineDGS Corpus

MeineDGS [67] is a German conversational sign language corpus which we used for the evaluation of our neural embedding based gloss search model from Chapter

5. The corpus consists of everyday conversations without a topic limitation, signed by 255 signers from 13 different regions across Germany to cover regional differences in signing. For the sign language utterances, the corpus have human pose estimates extracted using OpenPose [47] as well as RGB videos. The gloss annotations are given at both sentence level, i.e. with temporal sentence boundaries, and gloss level, i.e. with exact start and end times for each gloss, with type and sub-type hierarchies. The type glosses are related to the *iconic* value whereas sub-type glosses convey the meaning. In other words, the sub-type glosses inherit the form from types and change the meaning with additional movements and/or facial expressions. Therefore the target units in our application is selected to be type glosses, since our focus is on the manual features.

Since MeineDGS corpus by itself does not have a retrieval procedure, we developed a challenging signer-independent keyword search task so that none of the signers in the dataset are shared between our train, dev, and test partitions. Our rule in dividing the dataset was to have at least one signer in all the partitions. Thus, we had 183 signers for train, 36 signers across all 13 regions for dev, and the remaining 36 signers across all 13 regions for test partitions. Our labeling and training procedure was similar to the one we used in RWTH-Phoenix-Weather 2014T dataset: we did not utilize the gloss boundaries and used weak labels during both training and test. We searched for 2069 glosses that are shared across train and test sets, and calculated the retrieval scores on this relatively large vocabulary.

6.2. Evaluation Metrics

For each keyword, we sorted utterances that give the highest prediction scores and used different information retrieval metrics to measure the quality of the keyword search performance. More formally, for a query q, we calculate precision recall values at an operating point as:

$$\text{Precision} = \frac{|\{\text{Retrieved}\} \cap \{\text{Relevant}\}|}{|\{\text{Retrieved}|\}} \tag{6.1}$$

$$\text{Recall} = \frac{|\{\text{Retrieved}\} \cap \{\text{Relevant}\}|}{|\{\text{Relevant}\}|} \tag{6.2}$$

and use precision-recall based retrieval metrics that are obtained at different operating points (e.g. by changing the threshold) in evaluating the performance of our information retrieval systems.

6.2.1. Term-averaged Precision-Recall Curve and the F1 Score

When precision and recall values associated with a threshold θ is averaged over different queries q, term-averaged precision-recall values are obtained for that threshold:

$$\text{Precision}(\theta) = \frac{1}{|Q|} \sum_{q \in Q} \text{Precision}(q, \theta) \tag{6.3}$$

$$\text{Recall}(\theta) = \frac{1}{|Q|} \sum_{q \in Q} \text{Recall}(q, \theta) \tag{6.4}$$

Thus, by sweeping through different θ thresholds, we obtain the term-averaged precision-recall curve that summarize the performance of the keyword search system. We also report the maximum of F1 scores summarizing the curve:

$$\text{maxF1} = \max_{\theta} \frac{2 \cdot \text{Precision}(\theta) \cdot \text{Recall}(\theta)}{\text{Precision}(\theta) + \text{Recall}(\theta)} \tag{6.5}$$

6.2.2. Mean Average Precision (mAP)

Average precision (AP) for a keyword q is defined as:

$$\text{AP} = \frac{1}{|N|} \sum_{n=1}^{|N|} \text{Precision@}n(q) \tag{6.6}$$

After AP scores for each keyword is obtained, mAP score is calculated by taking the mean of average precision scores so that all the keywords are equally important, no matter how frequent they are in the test set.

6.2.3. Precision at 10 (p@10)

p@10 is the mean of precision scores at first ten retrieved utterances. It is a common metric in information retrieval for historical reasons, however, if the keyword is seen only once in the test set and that utterance is retrieved correctly, we still get p@10 score of 10% for this keyword.

6.2.4. Precision at N (p@N)

p@N is the mean of precision scores at first N_{test} retrieved utterances where N_{test}, the number of positive utterances in the test set, is different for each keyword.

6.2.5. Normalized Discounted Cumulative Gain (nDCG)

nDGC is a measure of ranking quality normalized with the ideal possible ranking. It weights the first retrieved utterances more and the gain gets smaller once we move into higher ranks.

6.3. Query-by-Example (QbE) Search

In our Query-by-Example Search [8] (Chapter 3) experiments, we search for citation-form isolated signs from BosphorusSign [68] dataset inside continuous sign language videos of the HospiSign [66] dataset. For a pair comprised of an isolated sign and a continuous sign sentence, we first find the distance between them, and compare this distance to the other pairs. Then, we calculate mean average precision (mAP) scores by sorting these distances for each citation-form sequence and then by looking at the correct labels.

Since we have each sign class repeated by eight different signers in this dataset, we report our results under two different settings: Same-Signer (SS) search and Signer Independent (SI) search. In *Same-Signer* (SS) search, results are collected for each of the eight users and average is taken afterwards. Thus, we measure if we can spot the same sign inside a longer sequence. In the more challenging task of the *Signer Independent* (SI) search, all the pairwise distances are collected in the same pool and distances are sorted together. It is expected that the distance for the same signs to be lower that different signs performed by the same signer. Thus, in SI search, we also measure the method's retrieval success when the sign is performed by a different signer.

In this section, we measure the performance of Query-by-Example Search for Sign Language under different distance metrics and feature extraction methods; with before (in Section 6.3.1) and after (in Section 6.3.2) temporal trimming is applied. In summary, we found that 1) OpenPose pose estimates, when used together with cosine distance, have the best performance. Moreover, the highest signer independence is also achieved under this setting. 2) temporal trimming increases the retrieval performance in all our experiments, especially when used with Fast Hand Descriptors.

6.3.1. The Effect of Different Feature Extraction Techniques and Distance Metrics on QbE

VGG-16 convolutional network [42], OpenPose pose estimation framework [47], and Fast Hand Descriptors (FHD) [44] are used in the feature extraction process. When using a Dynamic Time Warping based approach, it is also important to have the distance metric coherent with the features used. Thus, we also experimented with Euclidean and cosine distances for each feature. Query-by-example search results using different feature extraction techniques and distance metrics are summarized in Table 6.1. It can be seen that cosine distance generally results in better retrieval performance when used with most features.

Table 6.1. Query-by-example search results using three different features and two distance metrics. SS - Same signer mAP (%), SI - Signer independent mAP (%)

Feature	**Distance metric**	**SS (%)**	**SI (%)**
VGG-16	cosine	10.0	4.2
VGG-16	Euclidean	9.7	4.1
OpenPose	cosine	**26.2**	**19.2**
OpenPose	Euclidean	25.5	17.9
FHD	cosine	21.3	10.9
FHD	Euclidean	18.0	9.7

The robustness of features to signer independence is another important measure when selecting a feature extraction method for query-by-example keyword search. For a good feature, it is expected that performance for signer independent search would not be too far off from the same signer search. Thus, we report our signer independent results together with same signer setting in Table 6.1. When we consider the fact that random predictions result in a mAP score of 3.4% in this dataset, we can clearly see that VGG-16 features perform very bad for the signer independent task (mAP score of only 4.2%). The difference between same signer (10.0%) and signer independent (4.2%) performance is very prominent for VGG-16 features, which are trained on ImageNet dataset with different image classes. Our results suggest that this type of direct transfer learning approach from image recognition domain is not powerful for sign language retrieval.

For fast hand descriptors (FHD), we found that the retrieval performance in the same signer task is good, but performance drop is considerable in the signer independent task (mAP score dropped from 21.3% to 10.9%). Features obtained with OpenPose skeleton pose estimation framework, however, not only perform the best among the three features in all metrics, but also are the most robust to signer change. With only a slight drop from 26.2% mAP score in same signer search to 19.2% in signer independent task, Openpose features are found to be promising for sign language keyword search.

Table 6.2. QbE search results after temporal trimming. OpenPose + FHD denotes a fusion approach after we simply concatenate the features together. SS - Same signer mAP (%), SI - Signer independent mAP (%)

Feature	**Distance metric**	**SS (%)**	**SI (%)**
OpenPose	cosine	53.2	**37.2**
OpenPose	Euclidean	51.6	35.2
FHD	cosine	**56.3**	29.5
FHD	Euclidean	43.2	21.7
OpenPose + FHD	cosine	28.3	20.0
OpenPose + FHD	Euclidean	25.0	16.8

6.3.2. QbE Results After Application of Temporal Trimming

In the BosphorusSign dataset, signers start and end signing with hands in idle position as in most sign language datasets. When isolated signs are searched without cropping these transitional parts of the video, the subsequence matching capability of dynamic time warping algorithm is affected. Due to this reason, we report our results after a temporal trimming procedure (see Section 3.2) is applied. By comparing Tables 6.1 and 6.2, we can see that retrieval scores are improved for both OpenPose and FHD features after trimming. The biggest jump in performance can be seen when FHD features are used with cosine distance: the mAP score increased from 21.3% to 56.3% in same signer search, and from 10.9% to 29.5% in signer independent search. In Table 6.2, we also report our results with concatenated features. We see that fusing OpenPose and FHD features did not improve the retrieval scores in query-by-example search.

6.4. Gloss Search

Gloss is the basic unit of sign language, analogous to the word in a spoken language. When the gloss sequence is known for a sign language video, it can be viewed as its direct transcription into the main units of the language. Thus, for gloss-level keyword search, we can know for certain that if a gloss is present in the label of a sign language sentence, it is also present inside the video (contrary to cross-lingual keyword search where we cannot make sure if the word in the translation appears always in the same form).

Results for gloss search are obtained with our models described in Chapter 5, Keyword Search with Neural Embeddings. The main dataset we used for our experiments is RWTH-Phoenix-Weather 2014T [34], however, we also give some results in MeineDGS dataset [67] to show the general applicability.

6.4.1. Initial Results using Spatio-Temporal Graph Convolutional Encoder

We obtained our first successful results with using ST-GCN [61] as the encoder of our neural embedding based KWS model. Our initial results from which we derived our general KWS structure are enlisted in Table 6.3 for RWTH-Phoenix-Weather 2014T dataset. The standard encoder structure from the original ST-GCN implementation [61] forms our baseline: 10 layers of stacked ST-GCNs with two pooling layers and temporal kernel size of nine. Throughout the experiments, we saw that reducing temporal kernel size at the encoder side enables the selection mechanism to focus on the relevant temporal window of the sequence, resulting in a great leap in performance. Since the number of graph nodes increases by introducing hands to the layout, increasing the adjacency parameter of the graph convolution (neighbors) also resulted in a better performance.

We observed that cosine similarity based scoring functions were better than their dot product counterparts which are being utilized as go-to structures in attention

Table 6.3. The effect of different architectures on mAP score (%). Temporal focusing stands for a smaller kernel in temporal convolution. Learnable weights are β and θ are the learnable weights in the scoring function.

Layout	**Encoder Structure**	**Score**	**mAP (%)**
Right hand	Standard	dot product	5.51
Right hand	Standard	cosine similarity	15.55
Both hands	Standard	cosine similarity	13.94
Right hand	Temporal focusing	$\beta cos^2 + \theta$	19.46
Right hand	2-neighbor graph + Temporal focus	$\beta cos^2 + \theta$	20.73
Right hand	3-neighbor graph + Temporal focus	$\beta cos^2 + \theta$	24.09
Both hands	3-neighbor graph + Temporal focus	$\beta cos^2 + \theta$	**27.51**

mechanisms. Furthermore, introducing simple learnable weight and bias to the scoring function increases the performance. In our best performing model, we found optimal scoring weight β and bias θ to be 16.3 and 7.8.

Note that the addition of the second hand reduces the mAP score without temporal focusing and more graph neighbors. Because all the signers in the dataset are right handed, we experimented with skeleton data in the form of pose and only right hand (34 joints), and pose with both hands (55 joints). We observed that the addition of the second hand results in mixed effect in the mAP score. However, with the help of temporal focusing and flexing the adjacency definition on the graph, we obtained our best results with the addition of the second hand.

6.4.2. Effect of Different Encoder Structures on Gloss Search Performance

Gloss search results with various handshape and pose based models are compared in Table 6.4. From the comparison of Pose1 and Pose2 models, we can see that using confidence scores of OpenPose keypoint estimations increase the retrieval performance in every metric. Both hand shape based gloss search models in Table 6.4 perform

Table 6.4. Gloss search results (in %, the higher the better) using different encoder structures. UB: upper body, RH: right hand, LH: left hand, conf: OpenPose confidence scores. $\gamma > 0.5$ denotes increasing reliance on the pose model.

Gloss Search Models	**mAP**	**p@10**	**p@N**	**nDCG**
Pose1 (UB + RH + LH; x, y, conf)	29.24	26.25	25.84	47.52
Pose2 (UB + RH + LH; x, y)	28.05	24.97	24.38	47.02
Pose3 (UB + RH; x, y, conf)	29.21	26.15	25.94	47.68
Pose4 (UB; x, y, conf)	22.80	21.45	19.95	43.15
Multitask	23.54	23.03	20.71	42.89
Multitask + Pose1, γ=0.54	32.22	27.98	27.66	50.08
DeepHand	24.93	23.65	22.27	43.86
DeepHand + Pose1, γ=0.58	32.78	27.88	28.67	50.02

better than Pose4, i.e. pose based gloss search with only upper body; but addition of right hand in other pose based models increase the retrieval performance drastically. Comparing the two hand shape based models, we found that DeepHand features, which are trained on the domain data, perform universally better than Multitask features.

The results with the fusion of handshape based and pose based KWS model are also summarized in Table 6.4. We applied a late fusion approach described in Section 5.5 with γ values learned from development set. We see that using fusion of handshape based features and Pose1, we can surpass the recent gloss search performance. When we compare the fusion models, combining Pose1 with DeepHand is better than combination with the Multitask based one in many of the metrics. However, the Multitask features are trained with only out-of-domain data, and the difference between using Multitask features instead of DeepHand is minimal. Thus, we opted for combining Multitask with Pose1 as our go-to structure.

6.4.3. Gloss-Specific Comparison of Hand Shape and Pose Based Encoders

In this section, we show that some glosses can be retrieved more easily with handshape features whilst pose based KWS models are better for others. We qualitatively compare the model performances by looking six isolated sign samples. Since there is no ground truth labels in RWTH-PHOENIX-Weather 2014T, we use reference isolated signs taken from SignDict [1] German sign language dictionary for illustration. When selecting these six signs, we simply sorted all gloss queries according to the difference between Multitask and Pose1 models and picked the top three that also have a dictionary entry in SignDict for both extremes.

Table 6.5. Gloss-specific AP scores for different models. Both MultiTask and DeepHand features are extracted from right hand only. UB: upper body, RH: right hand, LH: left hand.

Gloss Search AP (%)	WENIG	BESSER	ELF	APRIL	GLEICH	NAH
Pose1 (UB+RH+LH)	7.47	17.22	19.24	85.24	76.39	50.81
Pose3 (UB+RH)	4.40	3.44	15.53	49.17	48.98	12.31
Pose4 (UB)	2.62	30.20	17.09	50.83	31.68	5.60
Multitask	55.06	100.00	74.34	8.12	1.48	2.40
DeepHand	62.94	81.25	60.51	3.52	13.83	3.32
Multitask + Pose1	43.10	75.00	36.12	67.19	61.48	45.65

Figure 6.1. Hand-picked definitive single frames for the signs in Table 6.5. Frames are taken from isolated videos in SignDict dictionary [1].

In Table 6.5, we can see that for the signs WENIG, BESSER, and ELF, both Multitask and DeepHand handshape based KWS models perform better than pose

based ones. From the dictionary entries for these signs in Figure 6.1, we see that all three of these signs are single-handed and formed of simple hand shapes. The signs APRIL, GLEICH and NAH are the among the signs where Pose based models perform significantly better than handshape based ones. When we do some qualitative analysis, we see that places of articulation are more important in defining these signs. In Figure 6.1, the sign for APRIL includes the thumb touching the nose and for GLEICH and NAH, we see hands interacting with each other. In Table 6.5, it can be seen that both Multitask and DeepHand handshape based encoders performed poorly compared to Pose1 model that includes upper body and both hands in the graph layout. Lastly, by observing the average performance in all these signs, we conclude that our Multitask + Pose1 fusion model performs reasonably better than relying on either hand shape or pose based models individually.

6.4.4. Analysis of the Fusion Model

The performance of Multitask + Pose1 model on different gloss vocabulary subsets are shown in Figure 6.2. When using weak labels during training, a single utterance is usually not enough to learn which temporal region is relevant for the sign. Thus, we also report our results in smaller vocabulary subsets. For 168 glosses with number of training samples $N_{train} \geqslant 50$, the mAP score is over 55%. For 115 glosses with $N_{train} \geqslant 100$, more than seven out of ten first retrieved utterances are correct. The results in Figure 6.2 follow a linear fashion other than the sharp increase in precision@10 scores. It is needed to have at least 10 positive utterances in the test set and this is true for most signs with $N_{train} \geqslant 100$.

6.4.5. Gloss Search Results on MeineDGS Corpus and Comparison with Phoenix-RWTH-Weather 2014T

In our experiments using MeineDGS corpus, we searched for 2069 in-vocabulary glosses using our pose-based KWS methods. The results with three different pose layouts can be seen in Table 6.6. In general, we see that the results on this dataset

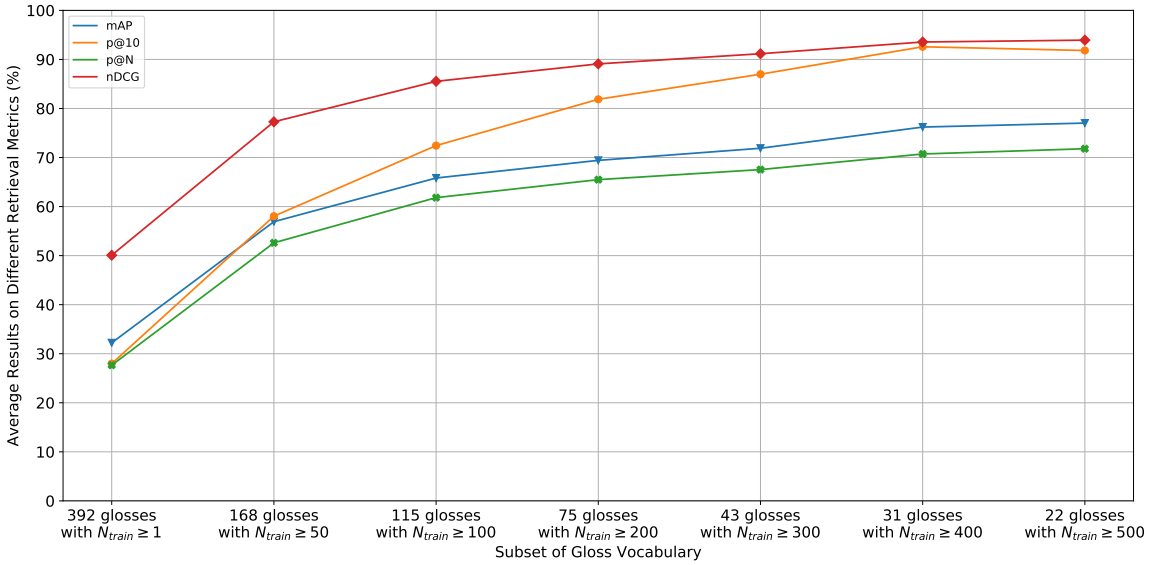

Figure 6.2. Gloss search results of the MultiTask+Pose1 fusion model on different vocabulary subsets. N_{train} denotes the number of training utterances positively labeled for that keyword.

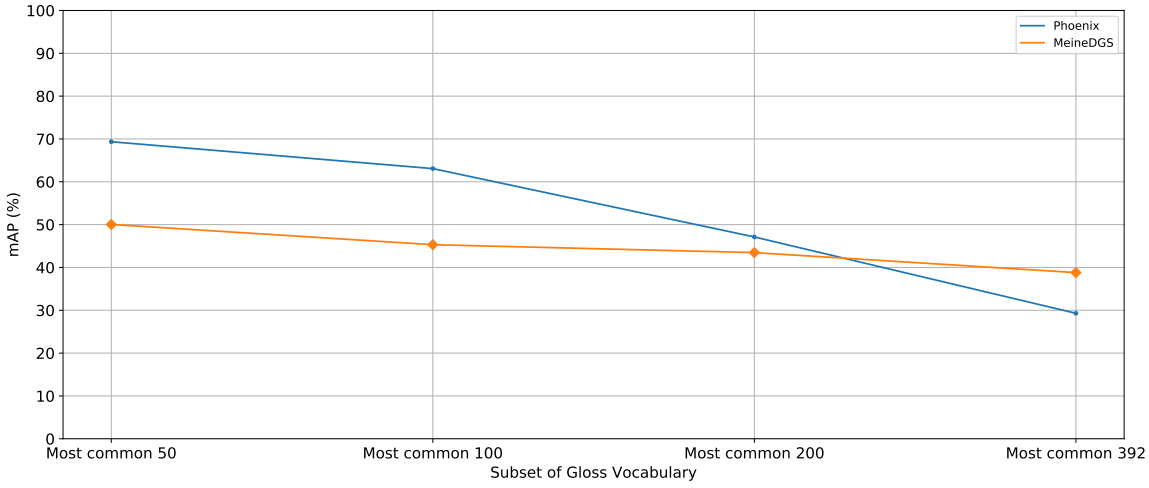

Figure 6.3. Gloss search results of the Pose1 model for most common words in RWTH-Phoenix-Weather 2014-T and MeineDGS datasets.

Table 6.6. Gloss search results on MeineDGS dataset (in %, the higher the better). UB: upper body, RH: right hand, LH: left hand, and conf refers to the use of OpenPose confidence scores alongside (x, y) spatial locations.

Gloss Search Models	**mAP**	**p@10**	**p@N**	**nDCG**
Pose1 (UB + RH + LH; x, y, conf)	17.85	16.63	16.66	37.22
Pose2 (UB + RH + LH; x, y)	14.42	14.40	14.12	33.58
Pose3 (UB + RH; x, y, conf)	14.91	15.22	14.74	34.06

follow a similar pattern to the ones in RWTH-Phoenix-Weather 2014-T. To compare the gloss search performance in two datasets, we also report our results in smaller vocabulary subsets since there is a mismatch in number of vocabulary elements between two datasets (392 glosses in Phoenix compared with 2069 glossses in MeineDGS). From the retrieval performance in different subsets in Figure 6.3, we see that KWS performance is better for the MeineDGS corpus in bigger vocabulary subsets whilst it is lower for smaller subsets.

6.5. Cross-Lingual Keyword Search

Transcribing sign language video into the gloss sequence is generally found to be costly, but written language translations are easily accessible. Employing written language translations in keyword search would not only suggest a cost-effective alternative to using glosses, but also provide a direct medium for enabling the non-deaf to learn the sign language in context. In this section, we perform keyword search in sign language sequences with these written language translations as the keywords. Since the sign language glosses and written language keywords do not match one-to-one, labels in the cross-lingual keyword search task cannot be viewed as the ground truth. However, with our experimental results in this section, we show that it is possible to perform retrieval even under this noisy and weakly-supervised setting.

To illustrate with an example, when we have "horse riding" in the written language translation of a sign language sentence, we label the sign language utterance

as if it includes the word "horse", however, "horse riding" has its own special sign in Turkish sign language and the actual sign for "horse" is non-existent in that video. Another example is when we have different words in different languages. When we search for "pain" inside a sign language sentence which includes "headache" in its written language translation, we may misguidedly think that the sign for "pain" is not in the utterance, but in the sign language "headache" is performed using "head" and "pain" one after another. Although the vocabularies may differ as in the above-mentioned examples, having written language translations as the only labels is still better than having no labels at all and utilizing cross-lingual KWS helps us in learning the units of the sign language.

We use two methods described in this thesis for Cross-Lingual KWS: 1) keyword search from decoder beams of a neural machine translation network, described in Chapter 4; and 2) keyword search with learning keyword-specific neural embeddings, described in Chapter 5. We report our cross-lingual keyword search results in the RWTH-Phoenix-Weather 2014T [34] dataset.

6.5.1. Results for Cross-Lingual KWS with Machine Translation

Our method described in Chapter 4 can be applied on the decoder results of any neural machine translation model. The results we report in this section are obtained with our method applied to S2T translation model from Neural Sign Language Translation [34], by changing the beam size parameter of the decoder module and whether or not applying a further normalization procedure.

Results pre-normalization can be viewed in Figure 6.4. The separate points denote the best translations obtained with beam sizes $k \in \{3, 10, 50, 500\}$. As can be seen from the figure, it is not possible to draw a statistically meaningful conclusion from these points which all lie around 20% Precision - 5% Recall. The curves in the figure represent the operating points obtained by sweeping the decision threshold with our method for the same beam sizes. As beam size increases, it can be seen that the

highest possible recall values also increase. This is because we start to consider paths with smaller probabilities as beam sizes increase.

For the precision values, however, the reverse statement can be made. From Table 6.7, we can see that mAP scores decrease with increasing beam sizes. This can be explained by observing Figure 6.4: for smaller beam sizes, maximum possible recall value decreases (from 27.8% max-recall when $k = 500$ to 7.3% max-recall when $k = 3$). Thus, as beam size decreases, mAP scores are calculated over smaller number of samples where model is more certain about the decision. To eliminate this bias, we also look at maxF1 scores. From Table 6.7, we can see that maxF1 scores does not get affected by this phenomenon and maximized around $k = 20$. Another conclusion visible from Table 6.7 is that as we increase the beam size parameter of the decoder, the number of retrievable keywords increases sharply. Thus, the decision regarding which beam size to choose should depend on our preference of precision-recall trade-off: retrieving a bigger subset of keywords and achieving a higher recall is possible by increasing the beam size, but this also means sacrificing from the precision.

Results after beam normalization are summarized in Table 6.7. If applied after decoding with bigger beam sizes ($k \geq 10$), normalization increases model's success in both mAP and maxF1 scores; but for smaller beam sizes ($k < 10$), beam normalization is harmful for keyword search. The main takeaway from all the curves in Figure 6.5 is that normalization is harmful for retrieval precision for very small ($< 5\%$) recall rates. However, if recall values $>$ %5 is needed for the application, beam normalization increases the precision. It can be seen that normalization fits all the operating curves in Figure 6.5 into a common plateau that passes through the operating points obtained with best-translations.

6.5.2. Results for Cross-Lingual KWS with Neural Embeddings

Cross-lingual KWS results obtained by using the method described in Chapter 5 is given in Table 6.8. We can see that pose based keyword search is better than relying

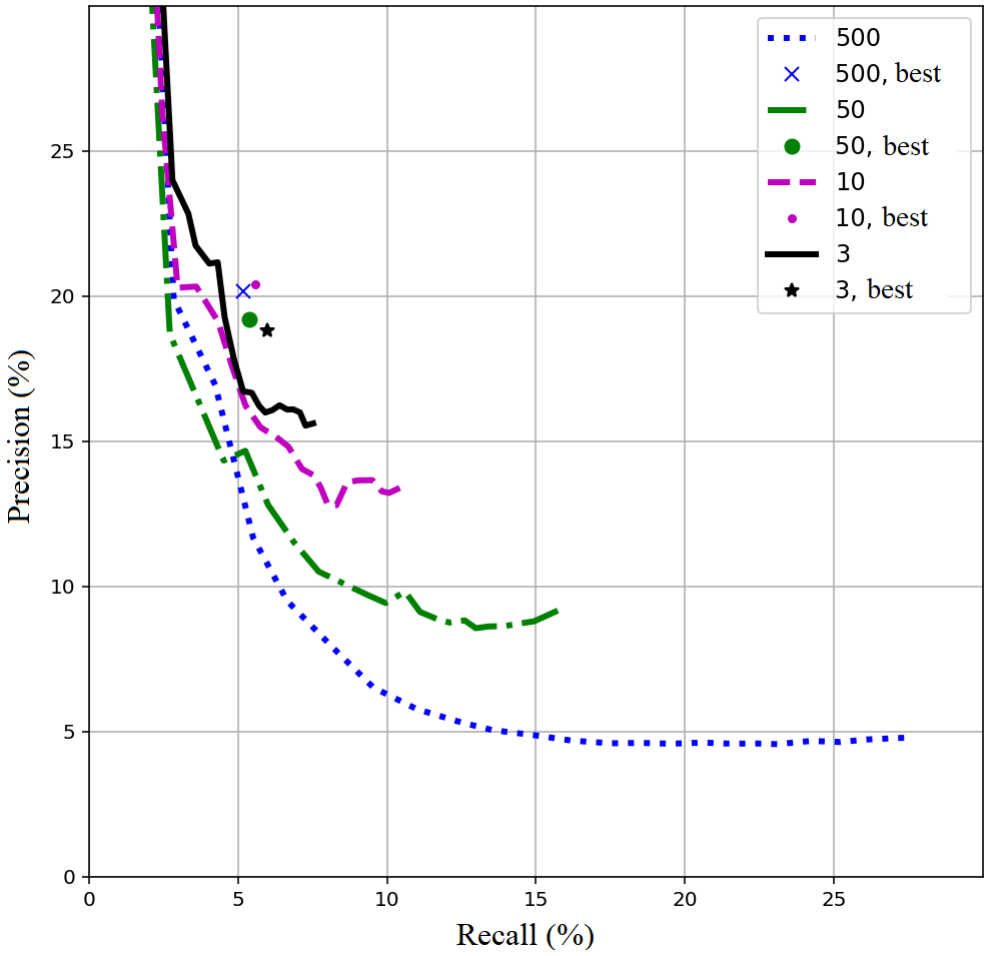

Figure 6.4. Pre-normalization Precision-Recall curves and best-translation operating points for different decoder beam sizes. Best means keyword search in the best translation, i.e. translation with the highest probability

on hand shape features in the encoder of the keyword search system. Among different pose models, the one incorporating both hands alongside the upper body performs the best. When we combine pose based and hand shape based keyword search models in a cold fusion approach, the retrieval performance increases.

6.5.3. Effect of Different Context Modeling Strategies

Results obtained with different context modeling strategies are summarized in Table 6.9. We see that statistical context modeling improves both metrics for all the layout options, and the gains are significant for the Upper Body + Both Hands

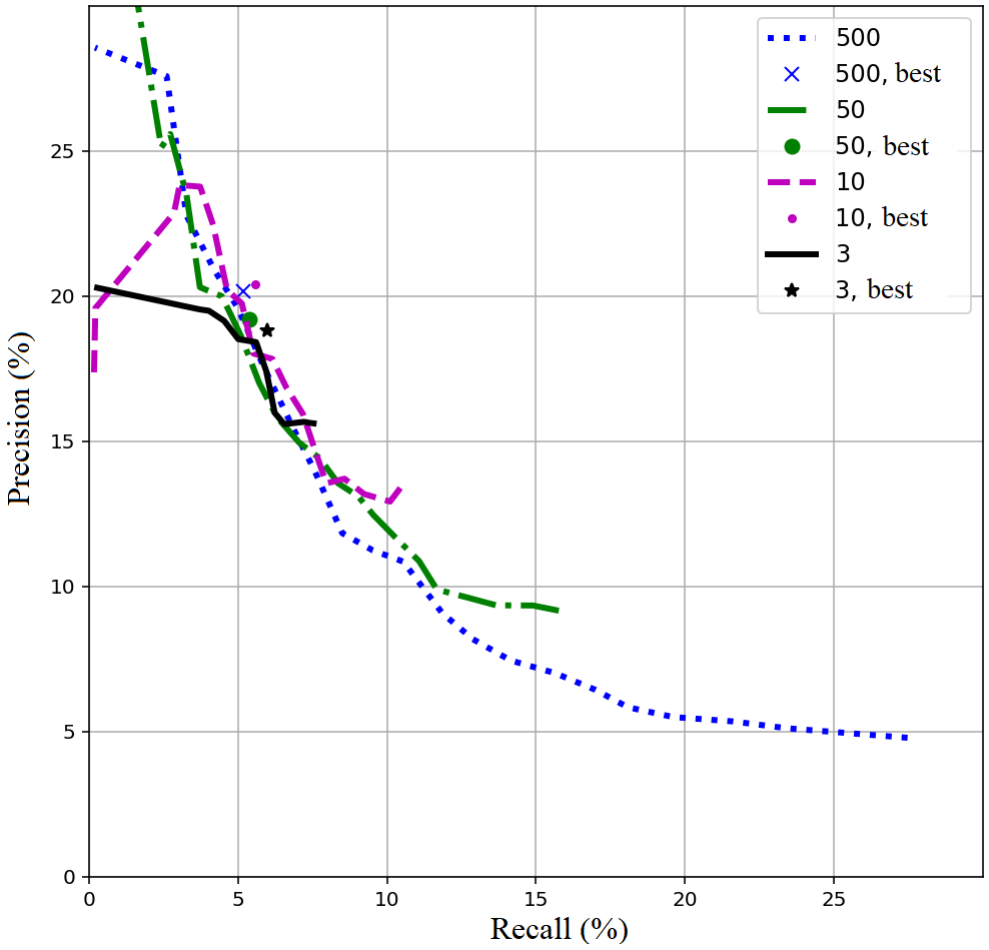

Figure 6.5. Precision-Recall curves and best-translation operating points for different decoder beam sizes after beam normalization. Best means keyword search in the best translation, i.e. translation with the highest probability

layout. Secondly, the MLP based context modeling did not improve the results for UB + Dominant Hand layout. Since we stopped the training of the MLP based context model when the development loss is not decreasing any further, the results in the test dataset are not necessarily better. However, we obtained our best overall mAP score with an MLP based context model (with a significant increase from 13.01% to 14.56%).

6.5.4. Comparison of Cross-Lingual KWS Methods

The best cross-lingual KWS results obtained with two techniques from Chapters 4 and 5 are compared in Figure 6.6. Since precision-recall curves from NSLT model with

Table 6.7. The effect of decoder beam size and beam normalization on mean average precision (mAP), maximum F1 score, and the number of retrievable unique words

Beam size	Pre-normalization		Normalized		Number of retrievable unique words
	mAP (%)	maxF1	mAP (%)	maxF1	
3	**44.71**	10.15	**41.8**	10.17	201
10	39.84	11.73	39.63	11.74	241
20	34.58	**12.45**	34.93	**12.44**	269
50	29.42	11.58	32.05	11.59	293
100	26.76	11.21	27.83	11.25	331
200	23.61	9.10	25.16	10.73	349
500	20.25	8.16	22.24	10.73	**392**

Table 6.8. Cross-lingual search results without context modeling (in %, the higher the better) using different encoder structures. UB: upper body, RH: right hand, LH: left hand, conf: OpenPose confidence scores. γ denotes the reliance on the pose model.

Cross-Lingual KWS Models	**mAP**	**p@10**	**p@N**	**nDCG**
Pose1 (UB + RH + LH; x, y, conf)	13.14	10.57	10.39	32.54
Pose2 (UB + RH + LH; x, y)	12.61	10.31	10.16	31.79
Pose3 (UB + RH; x, y, conf)	12.92	10.59	10.06	32.52
Pose4 (UB; x, y, conf)	10.79	8.77	8.73	29.99
Multitask	10.44	9.05	8.75	29.30
Multitask + Pose1, γ=0.68	14.34	11.52	11.27	33.66
DeepHand	11.11	9.62	9.14	29.85
DeepHand + Pose1, γ=0.60	14.75	11.43	11.63	33.97

different beam sizes all pass through a common plateau, we took the translation-based KWS results with the longest operating curve, i.e. with beam size 500. To illustrate the most successful cross-lingual model obtained with the techniques from Chapter 5, we took pose based keyword search model with the layout Pose1, and applied a MLP-based context modeling afterwards. And lastly, as the baseline, we searched for the

Table 6.9. Effect of context model (C.M.) on cross-lingual KWS. Best mAP and maxF1 scores for each layout are in bold. Pose1: Upper body and both hands, Pose3: Upper body and right hand, Pose4: Upper body only

	Without C. M.		**Statistical C.M.**		**MLP-based C. M.**	
	mAP (%)	maxF1	mAP (%)	maxF1	mAP (%)	maxF1
Pose1	13.01	16.40	13.80	**16.69**	**14.56**	16.15
Pose3	13.66	16.10	**13.78**	**16.28**	13.18	15.62
Pose4	11.62	14.66	11.94	14.90	**12.49**	**15.18**

keywords in NSLT [34] outputs and obtained a single operating point. By observing the term-averaged precision-recall curves from Figure 6.6, we concluded that the cross-lingual KWS model based on neural embeddings performs better than searching for keywords in translation outputs.

6.5.5. Temporal Localization as a By-Product of Weakly Supervised Training

When we have sequence-level, ordered gloss transcriptions of sign language data, HMM-based models can iteratively align each frame to a gloss hidden state and thus do the temporal segmentation as exemplified in [69]. However, since these HMM models rely on the strictness of the order of a gloss sequence, this alignment procedure cannot work with the noisy and weak supervision of translations. In this section, we show that our model's attention based selection mechanism can loosely localize some keywords independent of label type. For sign language sentences of varying length, we show the temporal keyword localization capabilities of our models that are trained with either gloss-sequences or translations as the labels.

In Figures 6.7, 6.8, and 6.9, we see model predictions (shown with percentages next to the labels) and related temporal localizations (denoted by the most peaky regions) for both gloss and cross-lingual search. For the most of our data, we see that

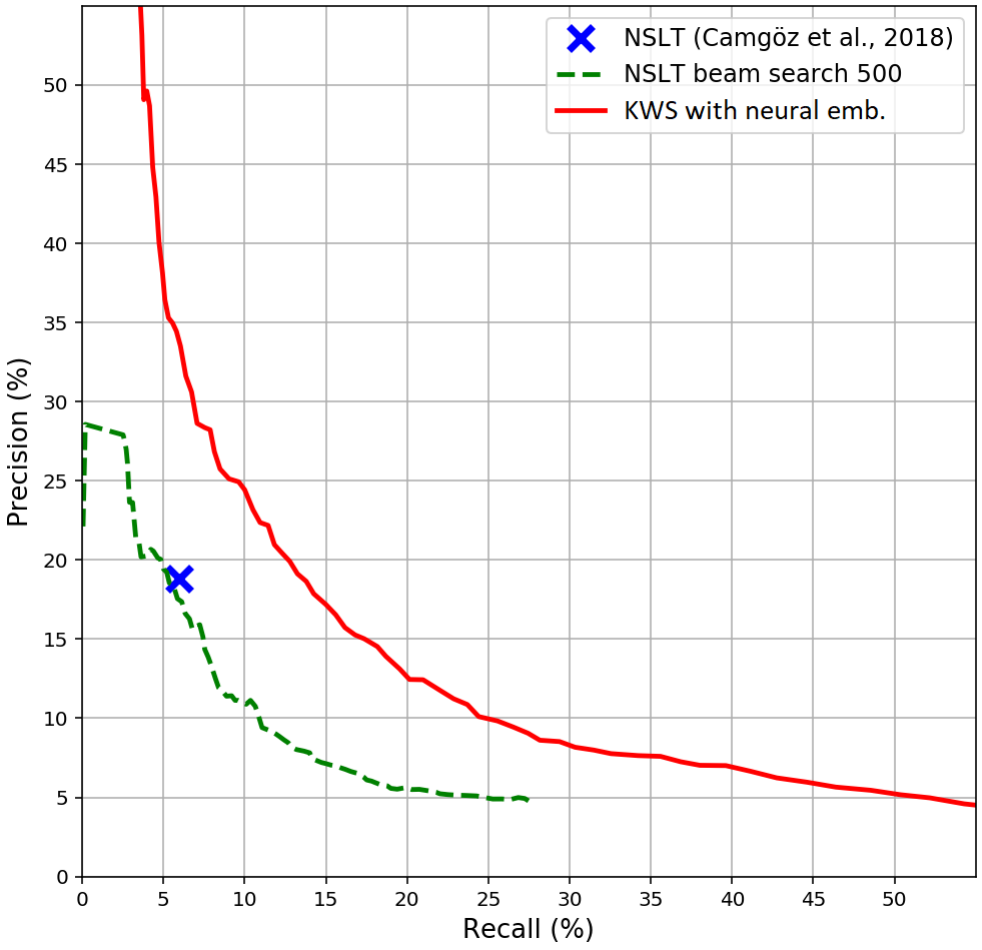

Figure 6.6. Our best cross-lingual KWS model (trained with Pose1 layout option and MLP context model) compared to searching from neural sign language translation outputs (the higher the better).

gloss search models are better in localization capacity and the order of peaky regions usually follows the gloss order correctly. We also see that peaky regions are more visible when the prediction confidences are higher. For the the cross-lingual search, we see that localization is possible for some words that are matching one-to-one with gloss transcriptions (such as “grad” in Figures 6.8 and 6.9, “sonntag” and “fünfundzwanzig” (with two peaks at both “fuenf” and “zwanzig”) in Figure 6.8, “nacht” in Figure 6.9 etc.), but not so much for the conjugated verbs like “bleibt” in Figure 6.7, or words without a unique gloss such as “alpenrand” and “ostseeküste” in Figure 6.9. We believe that cross-lingual KWS is at least beneficial for finding the most salient temporal regions that might be related to any gloss.

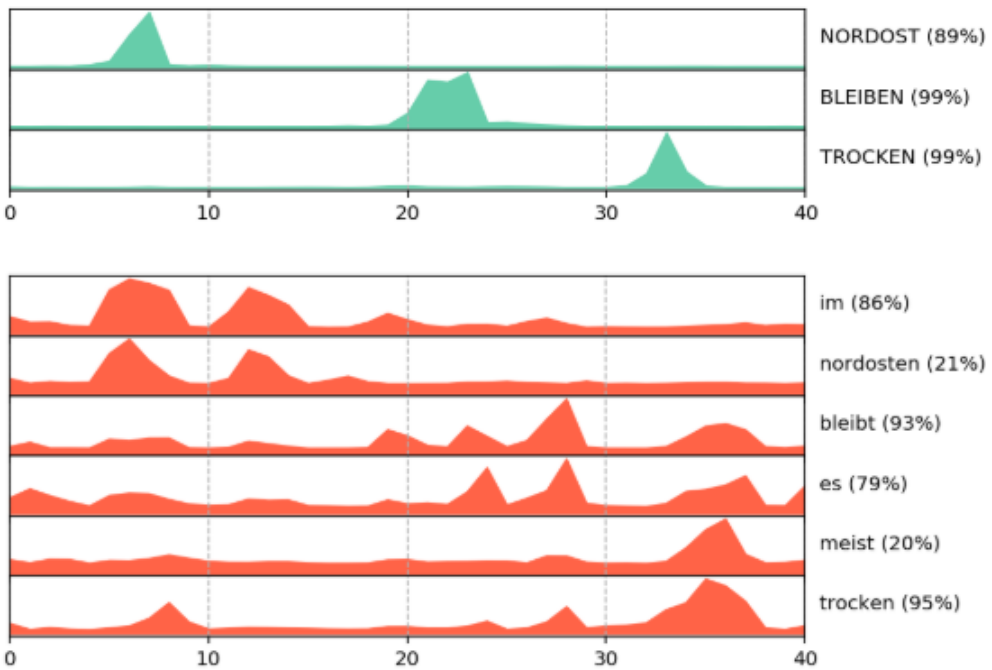

Figure 6.7. Temporal localizations for the sequence with gloss annotation "nordost bleiben trocken" and translation "im nordosten bleibt es meist trocken". The prediction confidences are denoted in parentheses.

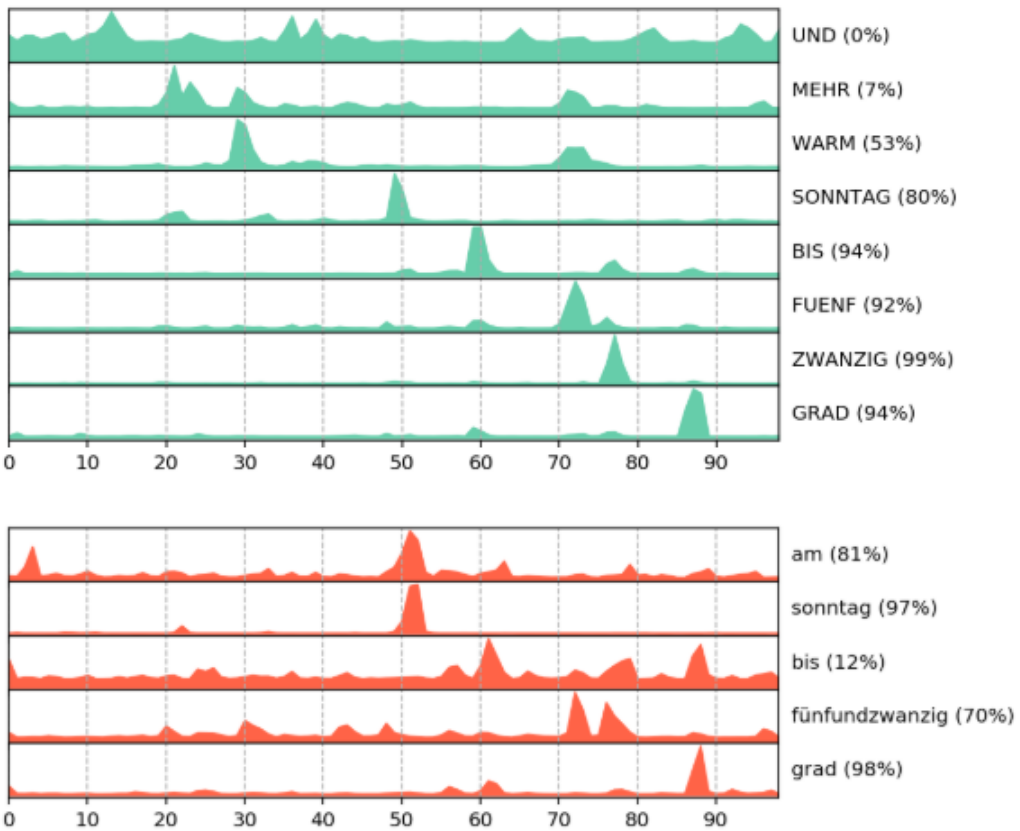

Figure 6.8. Temporal localizations for the sequence with gloss annotation “und mehr warm sonntag bis fuenf zwanzig grad” and translation “am sonntag bis fünfundzwanzig grad”. The prediction confidences are denoted in parentheses.

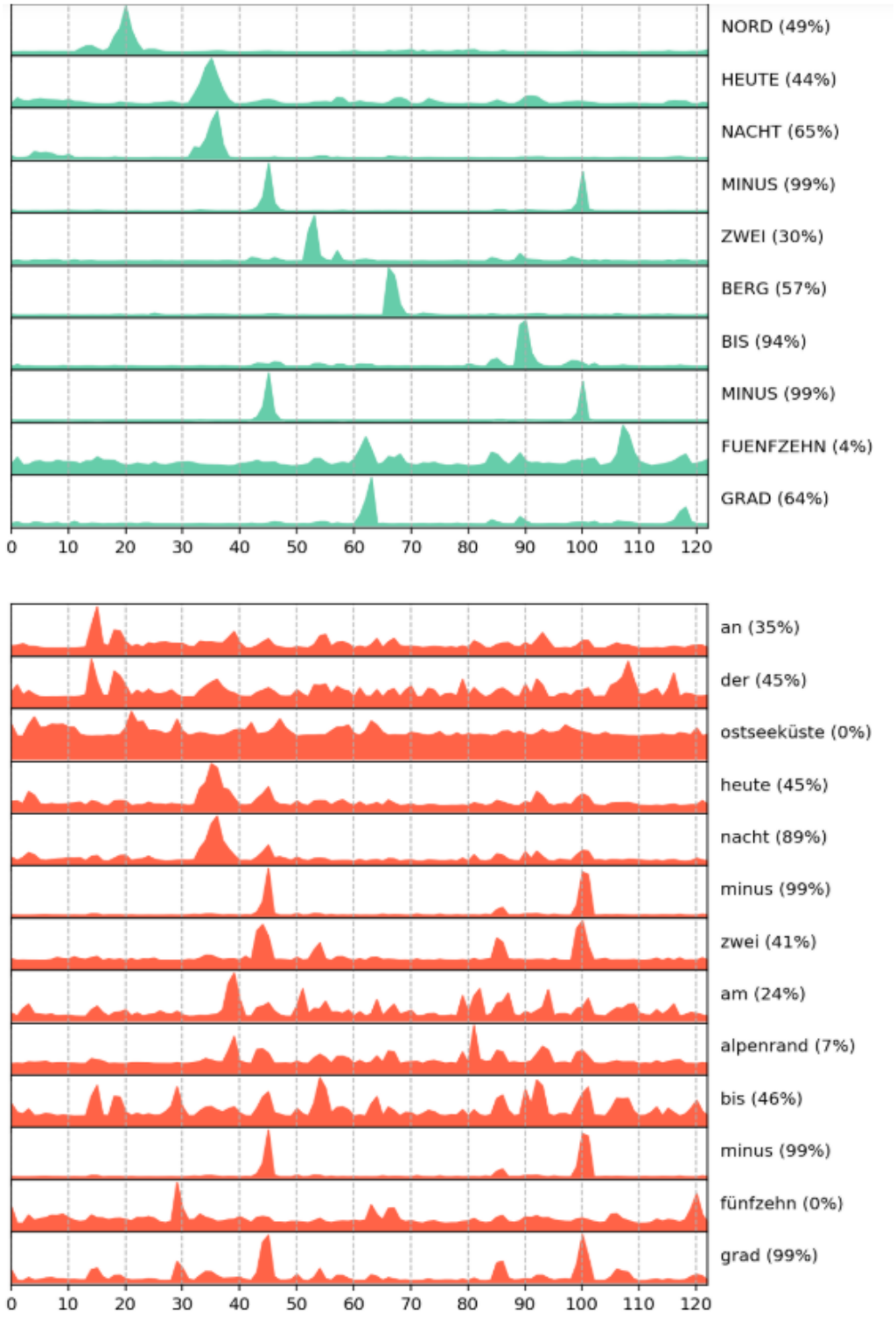

Figure 6.9. Temporal localizations for the sequence with gloss annotation “nord heute nacht minus zwei berg bis minus fuenfzehn grad” and translation ” “an der ostseeküste heute nacht minus zwei am alpenrand bis minus fünfzehn grad”.

7. CONCLUSION

In this thesis, we introduce the keyword search task for sign language and apply methods from several different domains to increase the retrieval performance in this new problem. We start by following similar procedures to current sign spotting algorithms in Query-by-Example search and show that human pose estimations obtained with OpenPose framework are robust features for sign retrieval. We then move into the domain of keyword search with written queries and develop methods which can be used with both gloss and cross-lingual keyword search. We use spatio-temporal graph convolutional networks together with human pose estimations on two different datasets and obtain our best performance. We then improve keyword search performance in sign language by incorporating hand shape features in a cold fusion approach. For the task of cross-lingual content retrieval from sign language, we first start by a more traditional approach based on outputs of a neural machine translation model, and then move into learning cross-lingual word embeddings. We further improve the cross-lingual keyword search performance by developing two different context modeling strategies.

Other than being an extensive study on information retrieval from sign language, we believe the methods we provide have potential use cases in the general problem of weakly supervised learning. Approaching the KWS problem with end-to-end paradigms that utilize weakly supervised learning is becoming more and more popular. However, to our knowledge, our use of attention together with keyword specific neural embeddings has not been attempted previously. With this novel approach from Chapter 5, searching for a large number of queries by focusing on different parts of the document for each query becomes possible and we observed in our experiments that such a temporal selection procedure increases the keyword search performance.

Another possible impact of this study can be seen in the field of cross lingual information retrieval. We are not aware of any previous cross-lingual keyword search strategy which relies on finding the expected counts across different translation paths

as we described in Chapter 4. We also developed another, less traditional technique for cross-lingual information retrieval from sign language with written language queries. The method we develop in Chapter 5 is also an important application of end-to-end techniques for cross-lingual information retrieval, and as we show, outperforms the translation-based technique. Although the method by itself does not include a language model, we incorporate a bag-of-words context modeling strategy to re-rank predictions for a keyword based on our method's own predictions for other keywords. We show that with this late language adaptation strategy, we can further increase the retrieval performance.

Currently, the KWS module from Section 5.3 makes decision about presence/ absence of the keyword based on a single context vector. This requires the model to look at the entire sequence during decision time, which hinders our ability to perform sub-sequence search. The most important future work of this thesis is to modify this network to incorporate a comparison procedure with the keyword embedding while generating the prediction from the context vector. With such a small difference, forming word lattices using this attention-based weakly supervised approach will become possible. Thus, we can find start and end boundaries of keywords with a greater temporal precision. After this, we can combine query-by-example with keyword search and generate word embeddings from example sign videos. Another research track which will become possible after this is translation-supervised term discovery: by training the sequence encoder with cross-lingual queries, it would be possible to find the basic building blocks of sign language.

REFERENCES

1. SignDict, "SignDict", `https://signdict.org/`, accessed in 27/08/2020.

2. Viitaniemi, V., T. Jantunen, L. Savolainen, M. Karppa and J. Laaksonen, "S-pot–a Benchmark in Spotting Signs within Continuous Signing", *Proc. Language Resources and Evaluation Conference (LREC), ISBN 978-2-9517408-8-4*, 2014.

3. Yang, H.-D., S. Sclaroff and S.-W. Lee, "Sign Language Spotting with a Threshold Model based on Conditional Random Fields", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 31, No. 7, pp. 1264–1277, 2008.

4. Cooper, H. and R. Bowden, "Learning Signs From Subtitles: A Weakly Supervised Approach to Sign Language Recognition", *Proc. Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2568–2574, IEEE, 2009.

5. Farhadi, A. and D. Forsyth, "Aligning ASL for Statistical Translation using a Discriminative Word Model", *Proc. Conference on Computer Vision and Pattern Recognition (CVPR)*, Vol. 2, pp. 1471–1476, IEEE, 2006.

6. Buehler, P., A. Zisserman and M. Everingham, "Learning Sign Language by Watching TV (using Weakly Aligned Subtitles)", *Proc. Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2961–2968, 2009.

7. Kelly, D., J. Mc Donald and C. Markham, "Weakly Supervised Training of a Sign Language Recognition System Using Multiple Instance Learning Density Matrices", *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, Vol. 41, No. 2, pp. 526–541, 2011.

8. Tamer, N. C., O. Özdemir, M. Saraçlar and L. Akarun, "Dynamic Time Warping Based Sign Retrieval", *Signal Processing and Communications Applications Conference (SIU)*, pp. 1–4, IEEE, 2019.

9. Tamer, N. C. and M. Saraçlar, "Keyword Search for Sign Language using Machine Translation", *Signal Processing and Communications Applications Conference (SIU)*, pp. 1–4, IEEE, 2020.

10. Tamer, N. C. and M. Saraçlar, "Keyword Search for Sign Language", *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 8184–8188, IEEE, 2020.

11. Tamer, N. C. and M. Saraçlar, "Cross-Lingual Keyword Search for Sign Language", *Proc. Language Resources and Evoluation Conference (LREC) Workshop on the Representation and Processing of Sign Languages*, pp. 217–223, 2020.

12. Tamer, N. C. and M. Saraçlar, "Improving Keyword Search Performance in Sign Language with Hand Shape Features", *Proc. European Conference on Computer Vision (ECCV) Workshop on Sign Language Recognition, Translation and Production*, pp. 1–4, IEEE, 2020.

13. Camgöz, N. C., A. A. Kindiroglu and L. Akarun, "Gesture Recognition using Template based Random Forest Classifiers", *Proc. European Conference on Computer Vision*, pp. 579–594, Springer, 2014.

14. Vaezi Joze, H. and O. Koller, "MS-ASL: A Large-Scale Data Set and Benchmark for Understanding American Sign Language", *Proc. British Machine Vision Conference*, 2019.

15. Özdemir, O., A. A. Kindiroglu and L. Akarun, "Isolated Sign Language Recognition with Fast Hand Descriptors", *Proc. Signal Processing and Communications Applications Conference (SIU)*, pp. 1–4, IEEE, 2018.

16. Zhang, J., W. Zhou, C. Xie, J. Pu and H. Li, "Chinese Sign Language Recognition with Adaptive HMM", *Proc. International Conference on Multimedia and Expo (ICME)*, pp. 1–6, IEEE, 2016.

17. Koller, O., O. Zargaran, H. Ney and R. Bowden, "Deep Sign: Hybrid CNN-HMM for Continuous Sign Language Recognition", *Proc. British Machine Vision Conference*, 2016.

18. Camgoz, N. C., S. Hadfield, O. Koller and R. Bowden, "Subunets: End-to-End Hand Shape and Continuous Sign Language Recognition", *Proc. International Conference on Computer Vision*, pp. 3075–3084, IEEE, 2017.

19. Cui, R., H. Liu and C. Zhang, "Recurrent Convolutional Neural Networks for Continuous Sign Language Recognition by Staged Optimization", *Proc. Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7361–7369, IEEE, 2017.

20. Saraclar, M. and R. Sproat, "Lattice-based Search for Spoken Utterance Retrieval", *Proc. Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL)*, pp. 129–136, 2004.

21. Chen, G., O. Yilmaz, J. Trmal, D. Povey and S. Khudanpur, "Using Proxies for OOV Keywords in the Keyword Search Task", *Proc. Workshop on Automatic Speech Recognition and Understanding*, pp. 416–421, IEEE, 2013.

22. Sarı, L., B. Gündoğdu and M. Saraçlar, "Fusion of LVCSR and posteriorgram based keyword search", *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

23. Audhkhasi, K., A. Rosenberg, A. Sethy, B. Ramabhadran and B. Kingsbury, "End-to-end ASR-free keyword search from speech", *IEEE Journal of Selected Topics in Signal Processing*, Vol. 11, No. 8, pp. 1351–1359, 2017.

24. Shan, C., J. Zhang, Y. Wang and L. Xie, "Attention-based End-to-End Models for Small-Footprint Keyword Spotting", *Proc. Interspeech*, pp. 2037–2041, 2018.

25. Park, A. S. and J. R. Glass, "Unsupervised pattern discovery in speech", *IEEE Transactions on Audio, Speech, and Language Processing*, Vol. 16, No. 1, pp. 186–

197, 2007.

26. Müller, M., "Dynamic Time Warping", *Information Retrieval for Music and Motion*, pp. 69–84, 2007.

27. Hazen, T. J., W. Shen and C. White, "Query-by-example spoken term detection using phonetic posteriorgram templates", *2009 IEEE Workshop on Automatic Speech Recognition & Understanding*, pp. 421–426, IEEE, 2009.

28. Ram, D., L. M. Werlen and H. Bourlard, "CNN Based Query by Example Spoken Term Detection.", *INTERSPEECH*, pp. 92–96, 2018.

29. Ram, D., L. Miculicich and H. Bourlard, "Neural Network based End-to-End Query by Example Spoken Term Detection", *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, Vol. 28, pp. 1416–1427, 2020.

30. Sutskever, I., O. Vinyals and Q. V. Le, "Sequence to Sequence Learning with Neural Networks", *Proc. Advances in Neural Information Processing Systems*, pp. 3104–3112, 2014.

31. Chung, J., C. Gulcehre, K. Cho and Y. Bengio, "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling", *arXiv preprint arXiv:1412.3555*, 2014.

32. Bahdanau, D., K. Cho and Y. Bengio, "Neural Machine Translation by Jointly Learning to Align and Translate", *arXiv preprint arXiv:1409.0473*, 2014.

33. Vaswani, A., N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser and I. Polosukhin, "Attention is All You Need", *Proc. Advances in Neural Information Processing Systems*, pp. 5998–6008, 2017.

34. Camgoz, C., S. Hadfield, O. Koller, H. Ney and R. Bowden, "Neural Sign Language Translation", *Proc. Conference on Computer Vision and Pattern Recognition*

(CVPR), pp. 7784–7793, 2018.

35. Ko, S.-K., C. J. Kim, H. Jung and C. Cho, "Neural Sign Language Translation based on Human Keypoint Estimation", *Applied Sciences*, Vol. 9, No. 13, p. 2683, 2019.

36. Orbay, A. and L. Akarun, "Neural Sign Language Translation by Learning Tokenization", *Proc. International Conference on Automatic Face and Gesture Recognition (FG)*, pp. 9–15, 2020.

37. Camgoz, N. C., O. Koller, S. Hadfield and R. Bowden, "Sign Language Transformers: Joint End-to-End Sign Language Recognition and Translation", *Proc. Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10023–10033, 2020.

38. Oard, D. W., "A Comparative Study of Query and Document Translation for Cross-Language Information Retrieval", *Proc. Conference of the Association for Machine Translation in the Americas*, pp. 472–483, Springer, 1998.

39. Saleh, S. and P. Pecina, "Reranking Hypotheses of Machine-Translated Queries for Cross-Lingual Information Retrieval", *Proc. International Conference of the Cross-Language Evaluation Forum for European Languages*, pp. 54–66, Springer, 2016.

40. Vulić, I. and M.-F. Moens, "Monolingual and Cross-Lingual Information Retrieval Models based on (Bilingual) Word Embeddings", *Proc. ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 363–372, 2015.

41. Zhang, L., D. Karakos, W. Hartmann, M. Srivastava, L. Tarlin, D. Akodes, S. K. Gouda, N. Bathool, L. Zhao, Z. Jiang *et al.*, "The 2019 BBN Cross-Lingual Information Retrieval System", *Proc. Workshop on Cross-Language Search and Summarization of Text and Speech (CLSSTS)*, pp. 44–51, 2020.

42. Simonyan, K. and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition", *Proc. International Conference on Learning Representations (ICLR)*, pp. 1–13, 2015.

43. Deng, J., W. Dong, R. Socher, L.-J. Li, K. Li and L. Fei-Fei, "Imagenet: A Large-Scale Hierarchical Image Database", *Proc. Conference on Computer Vision and Pattern Recognition*, pp. 248–255, IEEE, 2009.

44. Özdemir, O., A. A. Kindiroglu and L. Akarun, "Isolated Sign Language Recognition with Fast Hand Descriptors", *Signal Processing and Communications Applications Conference (SIU)*, pp. 1–4, IEEE, 2018.

45. Dalal, N. and B. Triggs, "Histograms of Oriented Gradients for Human Detection", *Proc. Conference on Computer Vision and Pattern Recognition (CVPR)*, Vol. 1, pp. 886–893, IEEE, 2005.

46. Poizner, H., U. Bellugi and V. Lutes-Driscoll, "Perception of American Sign Language in Dynamic Point-Light Displays.", *Journal of Experimental Psychology: Human Perception and Performance*, Vol. 7, No. 2, p. 430, 1981.

47. Cao, Z., T. Simon, S.-E. Wei and Y. Sheikh, "Realtime Multi-person 2D Pose Estimation Using Part Affinity Fields", *Proc. Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1302–1310, 2017.

48. Douglas, D. H. and T. K. Peucker, "Algorithms for the Reduction of the Number of Points Required to Represent a Digitized Line or its Caricature", *Cartographica: The International Journal for Geographic Information and Geovisualization*, Vol. 10, No. 2, pp. 112–122, 1973.

49. Mikolov, T., K. Chen, G. S. Corrado and J. Dean, "Efficient Estimation of Word Representations in Vector Space", *CoRR*, Vol. abs/1301.3781, 2013.

50. Krizhevsky, A., I. Sutskever and G. E. Hinton, "Imagenet Classification with Deep

Convolutional Neural Networks", *Proc. Advances in Neural Information Processing Systems*, pp. 1097–1105, 2012.

51. Hochreiter, S. and J. Schmidhuber, "Long Short-Term Memory", *Neural Computation*, Vol. 9, No. 8, pp. 1735–1780, 1997.

52. Luong, T., H. Pham and C. D. Manning, "Effective Approaches to Attention-based Neural Machine Translation.", *Proc. Conference on Empirical Methods in Natural Language Processing*, pp. 1412–1421, 2015.

53. Audhkhasi, K., A. Rosenberg, A. Sethy, B. Ramabhadran and B. Kingsbury, "End-to-end ASR-free keyword search from speech", *IEEE Journal of Selected Topics in Signal Processing*, Vol. 11, No. 8, pp. 1351–1359, 2017.

54. Koller, O., H. Ney and R. Bowden, "Deep Hand: How to Train a CNN on 1 Million Hand Images When Your Data Is Continuous and Weakly Labelled", *Proc. Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3793–3802, Jun. 2016.

55. Camgoz, N. C., S. Hadfield, O. Koller and R. Bowden, "Subunets: End-to-End Hand Shape and Continuous Sign Language Recognition", *Proc. International Conference on Computer Vision (ICCV)*, pp. 3075–3084, IEEE, 2017.

56. Gattupalli, S., A. Ghaderi and V. Athitsos, "Evaluation of Deep Learning based Pose Estimation for Sign Language Recognition", *Proc. ACM International Conference on Pervasive Technologies Related to Assistive Environments*, pp. 1–7, 2016.

57. de Amorim, C. C., D. Macêdo and C. Zanchettin, "Spatial-Temporal Graph Convolutional Networks for Sign Language Recognition", *Proc. International Conference on Artificial Neural Networks*, Vol. 78, pp. 646–657, Springer, 2019.

58. Koller, O., H. Ney and R. Bowden, "Read My Lips: Continuous Signer Independent

Weakly Supervised Viseme Recognition", *Proc. European Conference on Computer Vision*, pp. 281–296, Springer, 2014.

59. Ari, I., A. Uyar and L. Akarun, "Facial Feature Tracking and Expression Recognition for Sign Language", *Proc. International Symposium on Computer and Information Sciences*, pp. 1–6, IEEE, 2008.

60. Siyli, R. D., "HospiSign : A Framewise Annotated Turkish Sign Language Dataset", `http://dogasiyli.com/hospisign/`, accessed in 27/08/2020.

61. Yan, S., Y. Xiong and D. Lin, "Spatial Temporal Graph Convolutional Networks for Skeleton-based Action Recognition", *Proc. AAAI Conference on Artificial Intelligence*, pp. 1–10, 2018.

62. Kipf, T. N. and M. Welling, "Semi-Supervised Classification with Graph Convolutional Networks", *Proc. International Conference on Learning Representations (ICLR)*, pp. 1–14, 2017.

63. Karakos, D., R. Schwartz, S. Tsakalidis, L. Zhang, S. Ranjan, T. Ng, R. Hsiao, G. Saikumar, I. Bulyko, L. Nguyen *et al.*, "Score Normalization and System Combination for Improved Keyword Spotting", *2013 IEEE Workshop on Automatic Speech Recognition and Understanding*, pp. 210–215, IEEE, 2013.

64. Richards, J., M. Ma and A. Rosenberg, "Using Word Burst Analysis to Rescore Keyword Search Candidates on Low-Resource Languages", *Proc. International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7824–7828, IEEE, 2014.

65. Ramos, J. *et al.*, "Using TF-IDF to Determine Word Relevance in Document Queries", *Proc. Instructional Conference on Machine Learning*, Vol. 242, pp. 133–142, 2003.

66. Süzgün, M. M., H. Özdemir, N. C. Camgöz, A. A. Kındıroğlu, D. Başaran, C. Togay

and L. Akarun, "HospiSign: An Interactive Sign Language Platform for Hearing Impaired", *Deniz Bilimleri ve Mühendisliği Dergisi*, Vol. 11, No. 3, 2015.

67. Konrad, R., T. Hanke, G. Langer, D. Blanck, J. Bleicken, I. Hofmann, O. Jeziorski, L. König, S. König, R. Nishio, A. Regen, U. Salden, S. Wagner and S. Worseck, "MEINE DGS – annotated. Public Corpus of German Sign Language, 2nd release", `https://doi.org/10.25592/dgs.corpus-2.0`, accessed in 27/08/2020.

68. Camgöz, N. C., A. A. Kındıroğlu, S. Karabüklü, M. Kelepir, A. S. Özsoy and L. Akarun, "BosphorusSign: A Turkish Sign Language Recognition Corpus in Health and Finance domains", *Proc. International Conference on Language Resources and Evaluation (LREC)*, pp. 1383–1388, 2016.

69. Koller, O., S. Zargaran and H. Ney, "Re-Sign: Re-Aligned End-To-End Sequence Modelling With Deep Recurrent CNN-HMMs", *Proc. Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4297–4305, 2017.