Neural Sign Language Synthesis: Words Are Our Glosses

Jan Zelinka Jakub Kanis
University of West Bohemia, Faculty of Applied Sciences
New Technologies for the Information Society
Univerzitní 8, 306 14 Plzeň
{zelinka, jkanis}@ntis.zcu.cz

Abstract

This paper deals with a text-to-video sign language synthesis. Instead of direct video production, we focused on skeletal models production. Our main goal in this paper was to design a fully end-to-end automatic sign language synthesis system trained only on available free data (daily TV broadcasting). Thus, we excluded any manual video annotation. Furthermore, our designed approach even do not rely on any video segmentation. A proposed feed-forward transformer and recurrent transformer were investigated. To improve the performance of our sequence-to-sequence transformer, soft non-monotonic attention was employed in our training process. A benefit of character-level features was compared with word-level features. We focused our experiments on a weather forecasting dataset in the Czech Sign Language.

1. Introduction

This work builds on our previous research and applications [1, 2, 3, 4, 5] focused on Czech Sign Language (CSE). Our goal leads us to create a CSE corpora and design a sign language (SL) synthesis system to further advance the SL processing research area, especially for CSE.

Our main goal is to exclude any manual video annotation because any manual annotation is slow and costly and it is inconsistent more likely by its very nature. Moreover, SL speakers are far less accessible than the spoken language speakers. These problems make any method which relies on text or video annotations unsuitable or even impossible for TV broadcasting or some big-data-style source processing.

We utilized an internet archive of The Czech Daily News in CSE. Our experiments were focused on weather forecasting. This data source contains high definition videos with spoken commentary. However, there are no closed captions in this archive. Thus, we use texts obtained from the spoken commentaries by means of automatic speech recognition software which is able to convert this audio to text with high accuracy [6]. Instead of synthesizing of videos, we focused on skeletal models production that is in our opinion more versatile.

In this paper, we described two main tasks: to extract high-quality skeletal models from videos and to make fully trainable end-to-end SL synthesis system without any explicit translation. In our task, we don't have an alignment between spoken commentaries and relevant videos. Applied Dynamic Time Warping (DTW) or used non-monotonic attention replaces an alignment between input texts and sequences of skeletal models. But, especially in the case of bidirectional-RNN layers and the encoder-decoder approach, the alignment couldn't be derived because each word (or character) might affect each generated frame. Fortunately, we showed that the alignment is not necessary.

To extract quality skeletal models, we applied the OpenPose [7] – a third-party neural-network-based skeleton extraction method. Our skeletal models include head, arm joints, and all finger joints that are crucial for the SL understanding. However, finger joints are often misplaced and sometimes are even missing. These errors prevent using uncorrected skeletal models as ground truth. To correct used skeletal models extractor and to reconstruct some missing joints, we design a gradient-descend-based method for the skeletal models correction that creates 3D skeletal models from the extracted 2D models to arrange a geometrical consistency.

Our main contributions in this paper are the proposition of a simple but robust feed-forward translator, presentation of new criteria for the end-to-end system training and experiments with character-level features. The feed-forward translators replace RNN-based translators that are demanding whilst decrease the chosen error. The first investigated criterion is standard MSE with incorporated DTW. The second one is a designed criterion that uses soft non-monotonic attention instead of DTW. We found some benefits of a combination of the designed criteria.

2. Related Works

The classical approach to the solving of the problem of SL synthesis from spoken language is to divide it into the two main sub-tasks. There is a translation module which translates the spoken language (usually in a text form) to some text SL representation (rather machine readable), e.g. to glosses (lexical entities that represent individual signs). The second task is to render the SL animation video from the chosen SL representation. This animation is usually performed by an artificial computer avatar. The works [8] and [9] follow this classical approach. The recent notable works [10, 11, 5] push the state-of-the-art (sota) for SL processing forward by bringing the latest advanced techniques of Neural Machine Translation (NMT) and image generation to this research area.

The work [10] deals with the translation of the spoken language to the glosses by employing sota sequence-to-sequence (seq2seq) NMT approach based on the Recurrent Neural Network (RNN) with the attention mechanism. And subsequent direct generation of a sign video utterance from the given glosses (constituted by skeleton poses extracted from training data using OpenPose framework [7]) and basic speaker's pose using a method of direct image generation based on a convolutional image encoder followed by a Generative Adversarial Network (GAN). The second work [11] then covers the opposite direction of the translation from signs to words. The sign video is converted to spatial embeddings and then translated by the sota seq2seq NMT method to the words either using glosses as an intermediate representation or without it. The first mentioned approach achieves the best results.

Our solution to SL synthesis differs from [10] mainly by no utilizing of any SL translation/transcription or video annotation because we are using just raw SL video recordings. The output of our system is a sequence of skeleton poses which are, from our point of view, advantageous to a direct video generation because the final video animation can be performed by ordinary avatar. Additionally, in the case of skeletal models, we add the finger joints that are not considered in the work [10]. We also did not split the SL synthesis into the two sub-tasks solved separately, as it is in the classical manner, but we solve these sub-tasks jointly. From [11] we differ mainly in an investigation of the opposite direction of the direct translation between SL and spoken language and usage of skeleton poses instead of spatial embeddings for SL representation.

Our paper follows the paper [5] that is limited to RNN-based implicit translator, monotonic head and word-level features. We added feed-forward translators, new criteria and character-level features.

We proposed a backpropagation-based method to extract 3D skeletal models from 2D skeletal models obtained by OpenPose to correct errors and interpolate missing parts. We did not have any additional information about the speaker's bodies. Hence, we used a 3D model which was as simple as possible in contrast to another approach described in [12] than uses much more complex 3D deformation model. Other approaches for 3D body pose or hand pose estimation are described in [13, 14, 15, 16, 17, 18]. Because 3D estimation task wasn't our main goal, we chose to use the OpenPose framework as one of the feasible sota solutions.

In our solution, we use DTW to synchronize a resultant and a target sequence. The gradient propagated through DTW does not directly differentiate the optimal path. This disadvantage could be removed using so-called soft-DTW [19]. Due to the soft-DTW computational demands, we decided to investigate a replacement of the soft-DTW with an attention mechanism.

The proposed backpropagation-based correction method might be beneficially applied whatever pose estimator is (including 3D hand pose estimators) except for estimators such as [20] which keep results geometrical consistent. The 3D pose estimation itself wasn't our goal. However, a high-quality 3D pose and hand shape estimator such as estimators described in [13, 14] would allow us to construct a high-quality 3D sign language synthesizer.

3. Skeletal Model Extraction

In this section, we describe how we obtained our target data. Firstly, we briefly describe the results of used 2D skeletal models extraction and some problems with these results. Then we present our method which utilizes 3D skeletal models to provide some important corrections. After that, our method normalizing skeletal model size is explained. Finally, form of targets for machine learning is defined.

3.1. 2D Skeletal Extraction

We applied OpenPose [7] framework to extract poses from videos. The OpenPose works real-time and processes each picture in a video separately or with a short context. But a resultant poses are sometimes highly inaccurate (see Figure 1), some joints are even missing. Especially when a hand is not visible or it is blurred due to a rapid movement. Thus some corrections and interpolations are necessary to obtain useful target training data. We propose correction method that can process a whole video offline and utilize available context information.

3.2. 3D-model-based Skeletal Model Correction

Our solution is an iterative backpropagation-based algorithm. Some examples of the correction process are shown in Figure 2.

In order to make the corrections in a sequence of skeletal models, some invariants have to be found. Unfortunately, it

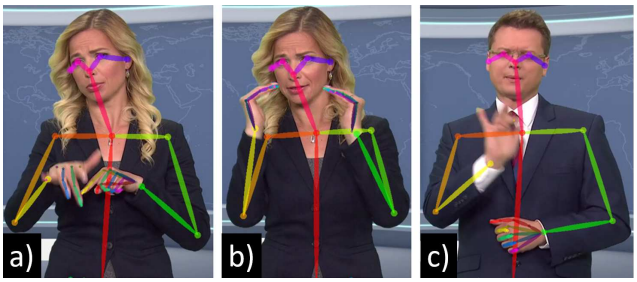

Figure 1. Examples of 2D skeletons produced by OpenPose framework. Despite the apparent high-quality skeleton extraction, some errors occurred: missing finger bones (a), covered fingers (b) and missing hand due to too rapid movement (c).

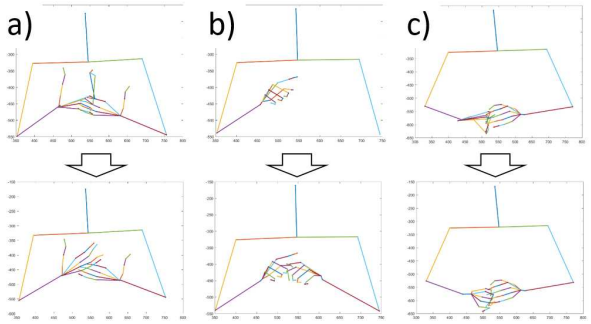

Figure 2. Examples of the correction process: misplaced joints correction (a, c) and missing bones correction (b).

is not easy to find helpful invariants in 2D skeletal models. On the other hand, in 3D skeletal models, obvious invariants such as bone lengths suggest itself.

3D model of a scene is usually acquired from two or more different points of view of the same scene [21, 22]. But we have only one shot of a scene but many shots from different times when bone lengths might remain constant. We not only considered constant bone lengths in each sequence but we also tied some bone lengths to make skeletal models strictly symmetrical (except angles).

Fortunately, obtained skeletal models have a tree structure with head as a root. This structure allows applying standard machine learning techniques for neural network training. The bone lengths, bone angles, and positions of a head are trainable parameters in our training process. The output of this "network" are joint positions. The loss function is Mean Squared Error (MSE) of joints 2D projections. Because we disregarded an effect of a perspective, the 2D projections simply cut out the third coordinate.

Our unsuccessful preliminary experiments with fully random initialization show that a high-quality initialization is crucial for usable 3D model estimation. Our initialization method works as follow: 1) We fixed initial 3D coordinates of the head as the same as the coordinates of the target 2D skeletons with $z = 0$. 2) Natural estimation of bone lengths as the maximum of lengths in 2D space is suitable only when no errors occur. We rather use an average of 2D bone lengths for our initialization. 3) We know the position of the head now. Other joint positions are computed recursively where each computation find the minimum of the aforementioned loss analytically.

The analytical solution of the steps of the recursion: Let suppose that we know a position $p_0 = (x_0, y_0, z_0)$ of the first joint of a bone with length L. The problem needs to be solved now is finding the minimum $p_1 = (x_1, y_1, z_1)$ of the criterion $e = (x_1 - x_{tar})^2 + (y_1 - y_{tar})^2$, where x_{tar} and y_{tar} are target coordinates, under the condition $\|p_1 - p_0\|^2 - L^2 = 0$. We used the method of Lagrange multipliers to solve the problem but the solution could be presented in very simple way using some visualization instead of rigorous reasoning: One can see that this problem has one or two solutions. The problem has one solution when the length L is not long enough to reach the target coordinates (or it is precisely long enough to reach the coordinates), i.e. when $L' \leq L$ where $L' := \sqrt{(x_{tar} - x_0)^2 + (y_{tar} - y_0)^2}$. In this case,

$$x_1 = x_0 + \frac{L}{L'}(x_{tar} - x_0), \tag{1}$$

$$y_1 = y_0 + \frac{L}{L'}(y_{tar} - y_0), \tag{2}$$

$$z_1 = 0. \tag{3}$$

For $L > L'$ both solutions are $x_1 = x_{tar}$ and $y_1 = y_{tar}$. The third coordinate z_1 is a solution of following equation $(x_{tar} - x_0)^2 + (y_{tar} - y_0)^2 + (z_1 - z_0)^2 = L^2$. It is very easy to find solution now:

$$z_1 = z_0 \pm \sqrt{L^2 - (x_0 - x_{tar})^2 - (y_0 - y_{tar})^2}. \tag{4}$$

Because we don't know which solution is more admissible, we chose the smaller one.

One can see that zero value of the MSE loss function could be found. However, such solution is highly undesirable because it leads to absurd bone lengths. To prevent this scenario and because we also want to eliminate too rapid movement and correct possible swapping between the two aforementioned possible solutions of 2D to 3D mapping, we used a typical L2 regularization. We regularize absolute values of joints velocities in a video and bone lengths of the skeleton.

3.3. Skeletal Models Scaling

Although video recording is done professionally, small variations of figure sizes are always present because the same speaker does not stay in the same spot during every take and each person has naturally different proportions. To reduce these differences, we apply a positive scales $s_i > 0$ for every sequence of skeletal models $x^i =$

$(x_1^i, \ldots, x_{n_i}^i)$ where $i = 1, \ldots, n$ resulted in scaled sequences $y^i = (s_i x_1^i, \ldots, s_i x_{n_i}^i)$. The simplest way how to set scales is to find the minimum of the following criterion $\varepsilon(s_1, \ldots, s_n) = \frac{1}{n} \sum_{i=1}^{n} \|s_i \mu_i - \mu\|^2$, where $\mu_i = \frac{\sum_{j=1}^{n_i} x_j^i}{n_i}$ is a local average and $\mu = \frac{\sum_{i=1}^{n} \sum_{j=1}^{n_i} x_j^i}{\sum_{i=1}^{n} n_i}$ is the global average vector of joints in the skeletal model. In other words: we want to scale each sequence to make the local averages as close to the global average as possible. One can see that this problem has a simple analytical solution $s_i = (\mu_i \mu^{\mathrm{T}}) \|\mu_i\|^{-2}$ if $s_i > 0$. We trained the scales using gradient descent. To keep $s_i > 0$, we trained $\hat{s}_i$ instead and $s_i = exp(\hat{s}_i)$.

3.4. Ground Truth Choice

Skeletal models have so far a form of an ordered set of joint coordinates. Estimating directly coordinates might lead to more accurate estimation from MSE loss perspective. However, a natural mutual position of bones is much more important than lesser MSE loss for plausible SL synthesis.

Thus, our ground truths are vectors of the bones, i.e. differences of the coordinates of joints on both ends of each bone. We found that a center of the chest (or rather the bottom end of the neck) nearly does not move. Furthermore, if it does move, it is not a part of any sign and we want to compensate for this movement. For this reason, we do not include any absolute coordinates. This choice not only eliminates the entirely irrelevant absolute position of the speakers in a picture but it also reduces the dimension from 100 (coordinates of 50 joints) to 98. Figure 4 shows that the results of this approach are naturally formed skeletal models. Finally, we normalized the average standard deviation of all points using one single coefficient. This final normalization serves only for faster training and does not change any proportions and relations in the target vectors.

4. Sign Language Synthesis

Our task is to convert an input Czech text into a sequence of skeletal models representing corresponding SL utterance. Several specifics of this task put standard sequence-to-sequence transformers in an unfavorable situation: The noise in available corpus caused by relatively random sign lengths and speakers differences is not sufficiently compensated by the size of the corpus that is rather sparse. Moreover, utterances in the corpus are not segmented, i.e. some utterances include several sentences. Hence, we use manageable sequence-to-sequence transformers.

Our SL synthesis has two parts. The first part is SL production that produces a sequence of skeletal models from a text and the second part is SL translation that translates text into inner implicit SL representation. Both parts are simplified: The SL production is a simplified repository with

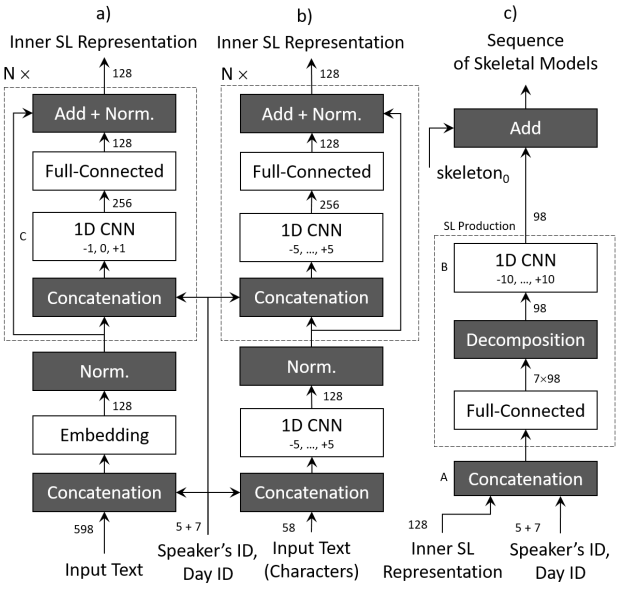

Figure 3. Schematic diagram of (a) the feed-forward model for a text-to-signs translation for word-level features, (b) the feed-forward model for a text-to-signs translation for character-level features, and (c) model for a sign-to-skeleton transformation.

constant length of sequences of skeletal models for each sign. The translator is simplified to produce an output text with the same length as an input text. These simplifications make the whole end-to-end system trainable.

4.1. Sign Language Production

In this section, we describe our designed technique that generates a sequence of skeletal models from word-level features obtained from spoken commentary. In addition to the spoken commentary, we also add information about speaker because the scaling described in Section 3.3 normalizes overall skeletal model sizes not differences in proportions and the normalization is not exact and some differences still remain. Furthermore, each speaker has not only different bone lengths but also each speaker uses distinctive signs and idioms. Some speakers even make some signs mirror-inverted. Additional information about speakers in one-hot codding helps our neural networks to improve the synthesis output quality.

We found another complication that speakers often add some information which is not included in a spoken commentary. Typically, a speaker uses the name of a day such as "on Monday" instead of the word "tomorrow". Hence, in addition to the speaker's identity, we add a vector that represents a day of the week. We didn't add any other additional information about an actual date to avoid revealing too much information relevant for a forecast.

Our simple sequence-to-sequence method replaces each input word with a short sequence of skeletal models with

constant length. We computed that each word in the spoken commentary is performed on average in $N = 7$ video frames in our training set, i.e. we divided all numbers of frames in videos by a number of words (tokens) in the training set. We use this number as the mentioned constant length of the short sequences. The method converts an input text $text = (word_1, \ldots, word_m)$ and the mentioned additional information into a sequence $x = [x_{i,j}]_{j=1,\ldots,N\cdot n}^{i=1,\ldots,m}$ where n is given number of features in a resultant sequence. The conversion is a simple trainable linear transform. To generate the resultant sequence, we designed a special layer that converts the sequence $x = [x_{i,j}]_{j=1,\ldots,Nn}^{i=1,\ldots,m}$ into the resultant sequence in the following way:

$$\begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,n} \\ x_{1,n+1} & x_{1,n+2} & \cdots & x_{1,2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{1,(N-1)n+1} & x_{1,(N-1)n+2} & \cdots & x_{1,Nn} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m,1} & x_{m,2} & \cdots & x_{m,n} \\ x_{m,n+1} & x_{m,n+2} & \cdots & x_{m,2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m,(N-1)n+1} & x_{m,(N-1)n+2} & \cdots & x_{m,Nn} \end{bmatrix} . \quad (5)$$

In other words, this transformation simply reshapes the input matrix by splitting up each row into N new subsequent rows. We call this operation as decomposition in Figure 3.

In the case of one-hot-codding (and no additional information), the short sequences are directly included in a weight matrix of the linear transform. This fact could be used for weight matrix initialization.

The special layer producing the resultant sequence is a linear and differentiable operation. This sequence production is a fully trainable repository. The main advantage of this sequence production is low computational demanding of gradient propagation. Especially in contrast to usual recurrent mechanism.

The evident disadvantage is that all signs have the same length. This disadvantage could be eliminated by DTW synchronization when some parts are repeatedly shortened and/or some parts are repeatedly lengthened. On the other hand, a well-trained preceding translator should split too long parts into two or more parts and omits irrelevant parts. A minor disadvantage is that this operation multiplies dimension N times. The main advantage is avoiding a recurrent layer.

Another characteristic of the proposed method is undesirable cuts on boundaries between words. Fortunately, this could be easily reduced employing a filter which smooths the resultant sequence. As such filter, we use a 1D Convolutional Neural Network (CNN) with a symmetric window. In our experiments, we used window with relative indexes $-10, -9, \ldots, +9, +10$.

An average of targets ($skeleton_0$) is added to the output to prevent a long initial phase of a model training. Figure 3 (c) shows the whole process of skeletal models production.

4.2. Sign Language Translation

Without appropriate data (parallel texts), we cannot explicitly train a complex translator but we train a system with a simplified structure that provides an implicit translation. The simplification lies in omitting the usual encoder-decoder passage and translating directly into a sentence with the same length. An SL differs from a spoken language not only on the lexical level but it has also different grammatical structure. Hence, a word-by-word translation is unusable. Our system translates words with their contexts using either 1D CNN or usual bidirectional GRU layers [23].

We use the additional information (speaker's ID) in our implicit translators because not only skeleton proportions and signs but also speaking manners and even grammar could be idiosyncratic.

The structure of our translator is shown in Figure 3 (a) and (b). The structure is similar to a structure described e.g. in [24]. In case of feed-forward translator, after a usual embedding layer, several blocks are applied. Each block consists of one 1D CNN with symmetrical window including the previous, the actual, and the next word. This layer uses ReLU activation function and a dropout (with dropout probability 0.1). The additional information is concatenated with an input of each block.

In the case of our RNN translator, the structure is the same. Only the CNN layer is replaced with a bidirectional GRU layer (see block noted as C in Figure 3 (a)).

5. Monotonic and Non-monotonic Attention

MSE can be computed when a resultant and a target sequence have the same length. But the sequences have different length and the sequences are probably not synchronized even if they have by accident the same length. Thus, we synchronize both sequences employing DTW or some attention mechanism. This section includes details of three proposed synchronization techniques: DTW, soft non-monotonic attention and their combination. Note that we strictly used MSE with DTW to evaluate our results in all experiments.

Because the used DTW could be seen as hard monotonic attention all three techniques use attention-based synchronization. We want to emphasize now that this attention is not an attention layer in our models, but we applied attention mechanisms in our loss. It means that our loss for a sequence $a = (a_1, \ldots, a_{n_a})$ and a sequence $b = (b_1, \ldots, b_{n_b})$

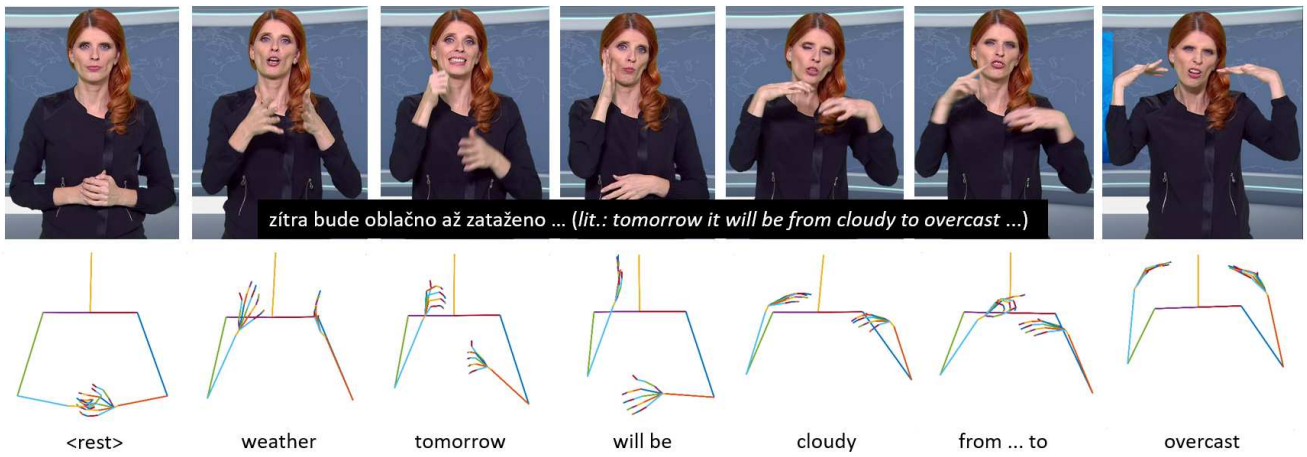

Figure 4. Examples of generated skeletons (generated by the most successful synthesis), real pictures from the test set, a relevant part of the commentary and manual description of the meanings of the signs.

is computed as follows:

$$\varepsilon = \frac{\sum_{i=1}^{n_a}\sum_{j=1}^{n_b} w_{i,j}\|a_i - b_j\|_D^2}{\sum_{i=1}^{n_a}\sum_{j=1}^{n_b} w_{i,j}}, \tag{6}$$

where $w(a,b) = [w_{i,j}]_{j=1,\dots,n_b}^{i=1,\dots,n_a}$ is an attention matrix and $\|\cdot\|_D^2$ is a chosen metric.

For DTW, $w_{i,j} = 1$ if a point (i,j) lies in the optimal path found by DTW algorithm, $w_{i,j} = 0$ otherwise and

$$\|a_i - b_j\|_D^2 = \sum_{k=-5}^{+5} \alpha_k \|a_{i+k} - b_{j+k}\|^2, \tag{7}$$

where $\alpha_k > 0$ are chosen weights. The purpose of this measure is to respect not only actual pose but to respect also whole local movement. We, therefore, use the symmetric window (the first or the last vectors are repeated instead of standard zero-padding that is unsuitable here). Because we chose to respect only short local movement, we heuristically chose $\alpha_k = \mathrm{softmax}(-0.1 \cdot k^2)$ for $k = -5, \dots, +5$. Another possibility is to employ standard delta and delta-delta acceleration coefficients.

A gradient propagated through any hard attention head is null everywhere heads yell null values. This characteristic might prevent a rectification of a poorly trained synthesizer. For this reason, we designed a soft head. An advantage of our soft non-monotonic head in comparison with monotonic heads is that it uses only a simple non-recurrent differentiable transformation of a distance matrix. A hard non-monotonic head has the same problem in gradient propagation as hard monotonic heads. Furthermore, a well-trained synthesizer leads to sharp head outputs. For these reasons, we preferred a soft non-monotonic head.

The main disadvantage of a monotonic head usage is that incorrect signs order might yell the same error as generating completely incorrect signs in this approach. To avoid this disadvantage, we design a soft and non-monotonic attention. Naturally, a target order of skeletal models should not be ignored completely. For this reason, the designed attention computes a matrix

$$w(a,b) = \mathrm{softmax}(-\hat{D}) + \mathrm{softmax}(-\hat{D}^{\mathrm{T}})^{\mathrm{T}}, \tag{8}$$

$$\hat{D} = M(n_a, n_b) \odot D(a,b), \tag{9}$$

$$D(a,b) = \left[\|a_i - b_j\|_D^2\right]_{j=1,\dots,n_b}^{i=1,\dots,n_a}, \tag{10}$$

where $\|a_i - b_j\|_D^2$ is the chosen measure described in (7), $\odot$ is element-wise product and matrix $M(n_a, n_b)$ is a mask given by the equation $M(n_a, n_b) = q_1 + q_2 \cdot d(n_a, n_b)$, where

$$d(n_a, n_b) = \left[\mathrm{e}^{-4\left(\frac{i}{n_a} - \frac{j}{n_b}\right)^2}\right]_{j=1,\dots,n_b}^{i=1,\dots,n_a}, \tag{11}$$

$q_1 = 32$, and $q_2 = -31$. Values of q_1 and q_2 were chosen to make values of $M(n_a, n_b)$ equal to one on diagonal and converge to 32 outside of diagonal. A purpose of the first softmax in (8) is to ensure that a model produces as many targets as possible and a purpose of the second softmax is to ensure that the model uses as many inputs as possible.

The correct signs order is preferred using the chosen metric $\|\cdot\|_D^2$. The mask M is chosen to make attention close to diagonal. This mask also helps to prefer correct order.

In our experiments, the hard monotonic attention (i.e. DTW) and soft non-monotonic attention were tested separately. Furthermore, a combination of these attentions (sum) was tested as well.

6. Word and Character Level Features

Using words as input has some disadvantages such as Out-Of-Vocabulary (OOV) words. Especially in the case of

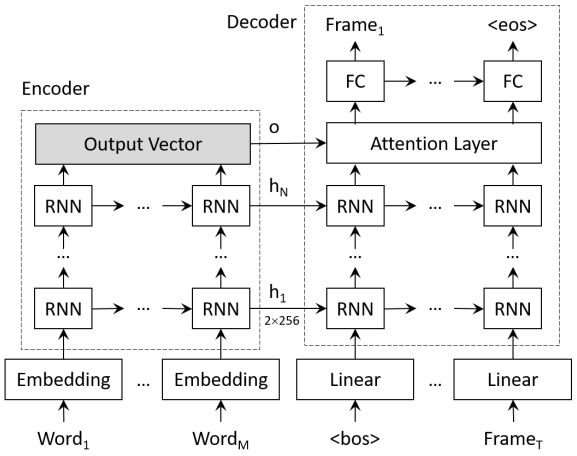

Figure 5. An overview of a seq2seq RNN-based translator.

the Czech language which is an inflectional language. Although we tried to eliminate this disadvantage using lemmatization, a similar problem with OOV words representing numbers remains. Furthermore, the used lemmatization naturally distinguishes between noun form, adjective form, verb form, and adverb form of the "same" word whilst CSE in some cases does not distinguish some of these forms.

Hence, it might be beneficial to use characters instead of whole words. Because numbers of characters are close to the numbers of frames in a video, special layers for words decomposition seems to be irrelevant. Naturally, models must be modified because using characters needs a much broader context. Thus, instead three-frames long symmetric window for word-level features, we used eleven-frames long symmetric window in our implicit translator. Instead of classical embedding layer, we also used 1D CNN with the eleven-frames long window. The modified translator for character-level features is shown in Figure 3 (b).

For character-level features, we slightly modified input texts of spoken commentaries: Each numeral written in digits was extended with zeros to have seven digits and special seven-characters long word was added on the beginning and on the end of each text. The special word consists of seven asterisks. (Asterisk was never used in any text up to now.)

7. Experiments and Results

We utilized an internet archive of The Czech TV news in CSE for our experiments. The Czech TV news in CSE is daily broadcast and videos are available online in high definition quality[1]. We focused only on weather forecasts processing. Each forecast video takes approximately half a minute. Our corpus contains 947 videos (from September 2015 to July 2018) of forecast in CSE performed by five different CSE speakers. 36 videos (cca 20,000 frames, cca 1500 words) are reserved for development tests, 36 videos are reserved for tests and the remaining 875 videos (cca 500,000 frames, cca 40,000 words) constitute our training dataset. To make our results more reproducible, we have performed each experiment three times with different train, development and test sets and weight initialization.

Day of the week of the broadcasting and speaker's IDs were included in videos and we transcribed this information manually from credits. We use the speaker's IDs in testing too. Alternatively, all possible speaker's IDs could be applied and the minimum could be found instead of using one known ID. All target 2D skeleton data were obtained from the source videos applying OpenPose and corrected by the designed method described in Section 3. Resulting targets have a dimensionality of 98 (49 bones in 2D space).

All automatically recognized Czech texts were lemmatized by MorphoDiTa [25] to decrease perplexity and to deal with OOV words. Our vocabulary contains 598 different Czech lemmas including a special symbol for the beginning and the end of the sentence. Both symbols are important not only for a translator but also for SL production because they correspond to resting skeleton pose on the beginning and the end of the video. Texts contain 58 different characters including space.

In our experiments, we measure square root of MSE to evaluate SL synthesis quality. Target vectors were normalized for the acceleration of our training process as we mentioned in Section 3.4. But for our evaluation, we used unnormalized targets. Values in the target vectors correspond to pixels in video frames. An overall result is an average of the results of the three sub-experiments. All results are in Table 1.

We adopted a technique described in [11] as a sota seq2seq translator that use an RNN-based encoder-decoder technique (with LSTM) that utilizes also an attention mechanism. The basic structure of our net remained the same as it is shown in Figure 5. We just reversed the translator model to model synthesis, i.e. we replaced CNNs and tokenization layer with a layer that performs word embedding and we replaced softmax activation function in the output layer of the decoder with a linear function. In our task, the sota translator faces these cardinal issues: the corpus contains a relatively small amount of data, videos are not segmented and most videos contain several sentences, and – unlike words in a text – each sign in a video takes a random amount of time. These issues probably make the errors of the encoder-decoder approach much higher than the errors of our more robust approach.

To make an ablative study, we firstly remove all dispensable parts. The first set of experiments was done without any translation. Input text was ordinary word-level features in one-hot coding. In the first experiment, we exclude CNN smoothing (block B in Figure 3) and do not include any ad-

[1] https://www.ceskatelevize.cz/ivysilani

ditional information, i.e. speaker's ID and day ID (block A in Figure 3). We investigated proposed non-monotonic soft attention, DTW-based hard monotonic attention and their combination. The benefit of CNN smoothing is investigated in the second experiment. The benefit of the additional information is investigated in the third experiment. The results show that the non-monotonic soft attention is not beneficial when it is used alone but it is significantly beneficial when it is combined with the DTW. In the next set of experiments, we, therefore, investigated only the DTW and the combination. The CNN smoothing and using additional information were significantly beneficial too and we used both in all next experiments. The lowest obtained error for the test set was 12.21.

To find out the limitations of the presented NN-based implicit translators, we designed a network that computes oracle annotations from target sequences. An oracle annotation is a result of a combination of two bottleneck techniques. The first one is a features bottleneck that reduces the dimension from 98 to 16. The second one is a time bottleneck that selects each 14th member of a sequence. To make the oracle annotations similar to the written text, we used the standard K-means where K equals to the number of unique words. An oracle synthesizer uses the oracle annotations. Its structure is shown in Figure 3 (c) and the results are in Table 1. The oracle sequences are representing sign language without any visible distortion. Some errors seem to be even corrections of pose estimation failure.

Table 1. MSE for proposed SL synthesis systems.

System	Criterion	Dev. Test	Test
Encoder-decoder	N/A	15.75	15.19
No imp. trans.,	Non-mono	14.91	14.34
no smoothing,	DTW	14.52	13.99
no add. inf.	Comb.	14.51	14.05
No imp. trans.,	Non-mono.	14.11	13.45
smoothing,	DTW	13.74	13.06
no add. inf.	Comb.	13.50	12.78
No imp. trans.,	Non-mono.	13.65	12.92
smoothing,	DTW	13.09	12.34
add. inf.	Comb.	**12.96**	**12.21**
Oracle	DTW	10.74	10.34
FF tr., N=4	DTW	12.70	11.93
FF tr., N=1	Comb.	**12.51**	**11.72**
RNN tr., N=4	DTW	12.73	11.98
RNN tr., N=4	Comb.	12.56	11.79
CLF, FF tr., N=1	DTW	12.77	12.07
CLF, FF tr., N=1	Comb.	**12.64**	**11.94**
CLF, RNN tr., N=1	DTW	13.28	12.59
CLF, RNN tr., N=1	Comb.	13.04	12.31

In the second set of experiments, an advantage of the proposed implicit translation is investigated. At first, feed-forward translation (noted as FF tr.) was investigated. We tried number of modules $N = 1, \ldots, 4$. After that, the proposed RNN-based translation (noted as RNN tr.) with bidirectional GRU was investigated. One can see that results for the feed-forward and the RNN translators are close. Nevertheless, feed-forward translator lowers error more. Furthermore, feed-forward translations trains and works much faster. The lowest obtained error for the test set was 11.72 (previous lowest error is 12.21).

The feed-forward and RNN translator were used in the last set of experiments. In these experiments, word-level features were replaced by character-level features (noted as CLF). The RNNs applied to character-level features have to process a much wider context. The feed-forward translators operate on a fixed sufficiently long context. The lowest obtained error for the test set was 11.94 that is increasing in comparison with the previous lowest error 11.72. The character-level features do not lead to faster computation in this case due to used translator which have to process approximately seven times longer sequences now. But, this approach could be more beneficial in the case of a corpus with a larger vocabulary.

8. Conclusion and Future Work

We presented our newly developed Czech SL synthesis that is a system that does not rely on any explicit SL translation neither in a training nor production process. We described a method for correcting skeleton poses and for interpolating missing skeletons parts. We present a special layer that allows producing a sequence of skeletons without any recurrent mechanism and could be used in machine learning. We designed a special feed-forward translator that could be trained simultaneously with our SL producer and that is suitable for a small corpus. We combined DTW and soft non-monotonic attention and investigated that this combination is beneficial. We also compared the word-level and the character-level features.

In our future work, we will utilize the whole Czech news in CSE and create several times larger corpus. We also plan to model facial expressions that convey crucial information in SL.

9. Acknowledgements

This work was supported by the European Regional Development Fund under the project AI&Reasoning (reg. no. CZ.02.1.01/0.0/0.0/15 003/0000466). Access to computing and storage facilities owned by parties and projects contributing to the National Grid Infrastructure MetaCentrum provided under the programme "Projects of Large Research, Development, and Innovations Infrastructures" (CESNET LM2015042), is greatly appreciated.

References

[1] Z. Krňoul, M. Železný, L. Müller, and J. Kanis, "Training of Coarticulation Models using Dominance Functions and Visual Unit Selection Methods for Audio-Visual Speech Synthesis," in *9th International Conference on Spoken Language Processing INTERSPEECH*, 2006, pp. 585–588.

[2] Z. Krňoul, J. Kanis, M. Železný, and L. Müller, "Czech text-to-sign speech synthesizer," in *4th International Workshop on Machine Learning for Multimodal Interaction*, 2008, pp. 180–191.

[3] J. Kanis, J. Zahradil, F. Jurčíček, and L. Müller, "Czech-Sign Speech corpus for semantic based machine translation," in *9th International Conference on Text, Speech and Dialogue TSD*, 2006, pp. 613–620.

[4] J. Kanis and L. Müller, "Advances in Czech - Signed Speech Translation," in *12th International Conference on Text, Speech and Dialogue TSD*, 2009, pp. 48–55.

[5] J. Zelinka, J. Kanis, and P. Salajka, "NN-based czech sign language synthesis," in *Speech and Computer - 21st International Conference, SPECOM 2019, Istanbul, Turkey, August 20-25, 2019, Proceedings*, 2019, pp. 559–568.

[6] J. Švec, M. Bulín, A. Pražák, and P. Ircing, "UWebASR - web-based ASR engine for Czech and Slovak," pp. 190–193, 2018.

[7] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, and Y. Sheikh, "OpenPose: realtime multi-person 2D pose estimation using Part Affinity Fields," in *arXiv preprint arXiv:1812.08008*, 2018.

[8] I. Almeida, L. Coheur, and S. Candeias, "Coupling natural language processing and animation synthesis in portuguese sign language translation," in *Proceedings of the Fourth Workshop on Vision and Language*. Lisbon, Portugal: Association for Computational Linguistics, Sep. 2015, pp. 94–103. [Online]. Available: https://www.aclweb.org/anthology/W15-2815

[9] L. Naert, C. Larboulette, and S. Gibet, "Coarticulation analysis for sign language synthesis," in *Universal Access in Human–Computer Interaction. Designing Novel Interactions*, M. Antona and C. Stephanidis, Eds. Cham: Springer International Publishing, 2017, pp. 55–75.

[10] S. Stoll, N. C. Camgöz, S. Hadfield, and R. Bowden, "Sign language production using neural machine translation and generative adversarial networks," in *BMVC*, 2018.

[11] N. C. Camgoz, S. Hadfield, O. Koller, H. Ney, and R. Bowden, "Neural sign language translation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[12] H. Joo, T. Simon, and Y. Sheikh, "Total capture: A 3D deformation model for tracking faces, hands, and bodies," *CoRR*, vol. abs/1801.01615, 2018.

[13] L. Ge, Z. Ren, Y. Li, Z. Xue, Y. Wang, J. Cai, and J. Yuan, "3D hand shape and pose estimation from a single RGB image," *CoRR*, vol. abs/1903.00812, 2019. [Online]. Available: http://arxiv.org/abs/1903.00812

[14] F. Mueller, F. Bernard, O. Sotnychenko, D. Mehta, S. Sridhar, D. Casas, and C. Theobalt, "GANerated hands for real-time 3D hand tracking from monocular RGB," in *Proceedings of Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[15] L. Chen, S.-Y. Lin, Y. Xie, H. Tang, Y. Xue, X. Xie, Y.-Y. Lin, and W. Fan, "TAGAN: Tonality aligned generative adversarial networks for realistic hand pose synthesis," in *British Machine Vision Conference BMVC*, 2019.

[16] X. Zhou, Q. Huang, X. Sun, X. Xue, and Y. Wei, "Weakly-supervised transfer for 3D human pose estimation in the wild," *CoRR*, vol. abs/1704.02447, 2017. [Online]. Available: http://arxiv.org/abs/1704.02447

[17] I.-C. Chang and S.-Y. Lin, "3D human motion tracking based on a progressive particle filter," *Pattern Recognition*, vol. 43, pp. 3621–3635, 2010.

[18] A. Arnab, C. Doersch, and A. Zisserman, "Exploiting temporal context for 3D human pose estimation in the wild," *CoRR*, vol. abs/1905.04266, 2019. [Online]. Available: http://arxiv.org/abs/1905.04266

[19] M. Cuturi and M. Blondel, "Soft-DTW: a differentiable loss function for time-series," in *Proceedings of the 34th International Conference on Machine Learning*, vol. 70. PMLR, 2017, pp. 894–903.

[20] Y. Luo, J. Ren, Z. Wang, W. Sun, J. Pan, J. Liu, J. Pang, and L. Lin, "LSTM pose machines," in *CVPR*, 2018.

[21] M. Kocabas, S. Karagoz, and E. Akbas, "Self-supervised learning of 3D human pose using multi-view geometry," *CoRR*, vol. abs/1903.02330, 2019. [Online]. Available: http://arxiv.org/abs/1903.02330

[22] T. Simon, H. Joo, I. A. Matthews, and Y. Sheikh, "Hand keypoint detection in single images using multiview bootstrapping," *CoRR*, vol. abs/1704.07809, 2017.

[23] R. Dey and F. M. Salemt, "Gate-variants of gated recurrent unit (GRU) neural networks," in *2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS)*, Aug 2017, pp. 1597–1600.

[24] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Curran Associates, Inc., 2017, pp. 5998–6008.

[25] J. Straková, M. Straka, and J. Hajič, "Open-Source Tools for Morphology, Lemmatization, POS Tagging and Named Entity Recognition," in *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. Baltimore, Maryland: Association for Computational Linguistics, June 2014, pp. 13–18.