

Gloss-Free End-to-End Sign Language Translation

Kezhou Lin¹ Xiaohan Wang¹ Linchao Zhu¹ Ke Sun² Bang Zhang² Yi Yang¹

¹ReLER, CCAI, Zhejiang University ²DAMO Academy, Alibaba Group

kezhoulin@zju.edu.cn wxh1996111@gmail.com zhulinchao7@gmail.com

xisheng.sk@alibaba-inc.com bangzhang@gmail.com yangyics@zju.edu.cn

Abstract

In this paper, we tackle the problem of sign language translation (SLT) without gloss annotations. Although intermediate representation like gloss has been proven effective, gloss annotations are hard to acquire, especially in large quantities. This limits the domain coverage of translation datasets, thus handicapping real-world applications. To mitigate this problem, we design the **Gloss-Free End-to-end** sign language translation framework (GloFE). Our method improves the performance of SLT in the gloss-free setting by exploiting the shared underlying semantics of signs and the corresponding spoken translation. Common concepts are extracted from the text and used as a weak form of intermediate representation. The global embedding of these concepts is used as a query for cross-attention to find the corresponding information within the learned visual features. In a contrastive manner, we encourage the similarity of query results between samples containing such concepts and decrease those that do not. We obtained state-of-the-art results on large-scale datasets, including OpenASL and How2Sign.¹

1 Introduction

Sign language is a type of visual language mainly used by the community of deaf and hard of hearing. It uses a combination of hand gestures, facial expressions, and body movements to convey the message of the signer. Sign languages are not simple transcripts of the corresponding spoken languages. They possess unique grammar structures and have their own linguistic properties. According to the World Federation of the Deaf, there are over 70 million deaf people around the world. The study of automated sign language processing can facilitate their day-to-day life.

In this paper, we study the task of sign language translation (SLT), which translates the sign videos

¹Our code and model will be available at <https://github.com/HenryLittle/GloFE>.

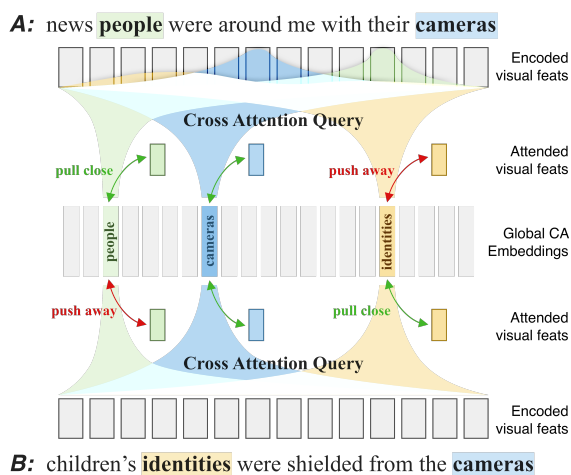


Figure 1: Use global embeddings of conceptual words (CA, conceptual anchor) in spoken translation to supervise the visual feature instead of gloss. **A** and **B** are different samples from the same mini batch.

into the corresponding spoken language. Glosses are the transliteration system for sign language. They serve as an intermediate representation of the signs. However, the vocabulary of gloss does not align with the spoken language nor does the order of the glosses. Unlike translation between two spoken languages, the number of frames in a sign video is much larger than the number of words in the spoken translation. This imposes a unique challenge for SLT. Models need to learn a clustering of the frames into gloss-level representation before they can translate the tokens. Previous methods solve this problem in two major ways, i.e., pre-train the visual backbone with gloss (Camgoz et al., 2020) or jointly train on both translation and continuous recognition task (Camgoz et al., 2020; Chen et al., 2022) with an additional CTC loss (Graves et al., 2006). These methods have been proven effective, but the reliance on gloss annotations makes them hard to apply to more realistic scenarios. As gloss annotations require expert knowledge to make and often are limited in

quantity or coverage of domains. Like the most frequently used PHOENIX14T dataset (Camgoz et al., 2018) that focuses on weather reports or the KETI dataset (Ko et al., 2019) that dedicates to emergencies. Datasets like OpenASL (Shi et al., 2022) and How2Sign (Duarte et al., 2021) provide more samples but there are no gloss annotations for training.

Motivated by these observations and the availability of large-scale SLT datasets, we designed a new framework that is gloss-free throughout the entire process and train the visual backbone jointly in an end-to-end manner. The core idea of our method is illustrated in Figure 1, we extract conceptual words from the ground truth spoken translation to be used as a weak form of intermediate representations. This exploits the shared semantics between signs and text. Though the extracted words might be different from the glosses, the concept expressed by these words should exist in both sign and text. We treat these words as conceptual anchors (CA) between the two modalities. Specifically, we use pre-trained GloVe embeddings (Pennington et al., 2014) as the initialization of these anchors. Then they are treated as the query of cross attention against the encoded visual features. As illustrated in Figure 1, the query attend to each visual feature across the temporal dimension to calculate the similarity between the query and the visual features. With these similarities as weights of pooling, we get the attended visual features. The order of the most relevant features from the signing video does not match the order of the queries in the translation, so CTC is not viable in this situation. Instead, we impose the conceptual constraints in a contrastive manner. For each anchor word, we treated samples containing such words as positive and vice versus. For example, for the word identities in Figure 1 sample B is positive and sample A is negative. Query results for these positive and negative pairs along with the anchor word form a triplet, among which we conduct a hinge-based triplet loss. This process forces the visual2text encoder to learn the relation between different frames that is part of one sign. In all, our contribution can be summarized as:

- An end-to-end sign language translation framework that takes the visual backbone in its training process. And we prove that proper design to accompany the text generation objective, will improve the performance of the

framework rather than deteriorate it.

- A replacement for gloss as a weak form of intermediate representation that facilitates the training of the visual backbone and encoder. It exploits the shared semantics between sign and text, bridging the gap between these two modalities. This also allows us to train the model on larger datasets without gloss annotations.
- We obtained state-of-the-art performance on the currently largest SLT dataset publicly available, improving the more modern BLEURT metric by a margin of 5.26, which is 16.9% higher than the previous state-of-the-art.

2 Related Work

2.1 Sign Language Translation

Sign Language Translation: Sign language translation (SLT) aims to translate a sign video containing multiple signs to the corresponding spoken text. Camgoz et al. (2018) first proposed the PHOENIX14T dataset that enables the study of direct translation from sign videos to spoken translation. Due to the data scarcity issues caused by the cost of labeling gloss in large quantities. Most works on SLT focus on exploit gloss annotations (Camgoz et al., 2020) or techniques like back translation (Zhou et al., 2021) between gloss and spoken text. Chen et al. (2022) transfers powerful pre-trained models (Radford et al., 2019; Liu et al., 2020a) to the sign domain through progressively pre-training and a mapper network. PET (Jin et al., 2022) utilizes the part-of-speech tag as prior knowledge to guide the text generation. However, they all rely on gloss annotations. There have been attempts to conduct SLT in a gloss-free manner (Camgoz et al., 2018; Li et al., 2020b; Kim et al., 2022), but their results are subpar compared to those that use gloss annotation. Recently, there have emerged large-scale SLT datasets like How2Sign (Duarte et al., 2021) and OpenASL (Shi et al., 2022). They both surpass PHOENIX14T in quantity and are not limited to a certain domain. However, these two datasets don't provide gloss annotations. By far, there are few frameworks have been developed to tackle this challenging scenario except for the baseline methods of the datasets.

2.2 Pretraining with Weakly Paired Data

Vision-language pretraining (Radford et al., 2021; Tan and Bansal, 2019; Chen et al., 2020) on massive-scale weakly paired image-text data has recently achieved rapid progress. It has been proven that transferable cross-modal representations bring significant gains on downstream tasks (Ri and Tsuruoka, 2022; Ling et al., 2022; Agrawal et al., 2022). Recent endeavors (Yu et al., 2022; Desai and Johnson, 2021; Wang et al., 2021; Seo et al., 2022) leverage generative pretraining tasks like captioning to enable the cross-modal generation capability. Such a training regime has become increasingly popular in sign language translation. In particular, a few early attempts (Kim et al., 2022) directly adopted the translation loss for cross-modal learning. However, the translation objective is hard to learn an effective representation of the important concept, especially in an open domain scenario. In contrast, we design a contrastive concept mining scheme to address this problem, leading to performance gains on the two largest sign language translation datasets.

3 Method

Given a sign video $X = \{f_1, f_2, \dots, f_T\}$ of T frames, our objective is to generate a spoken language sentence $Y = \{w_1, w_2, \dots, w_L\}$ of length L under the conditional probability $p(Y|X)$. Generally speaking, it holds that $T \gg L$. This trait makes the task of sign language translation harder compared to the translation task between different spoken languages. Past methods mostly use gloss supervision via CTC loss to impose an indirect clustering on the processed visual tokens. Gloss annotation provides the relative order and type of the signed word, not including the boundary between sign words. However, the making process of gloss annotations is labor-intensive, thus often in limited quantities. This restricts the scale of SLT datasets with gloss annotations.

To this end, we are motivated to design a framework that can be trained only on sign video and translation pairs. To reduce processing load and translate longer sign videos, we extract pose landmarks $X_{pose} = \{p_1, p_2, \dots, p_T\}$ offline from X and use it as the input of our framework. In this section, we first give an overview of the proposed gloss-free end-to-end sign language translation framework, with details about each component. Then we elaborate on our approach aims to provide

similar supervision to gloss in a self-supervised manner.

3.1 Framework Overview

The overall structure of our framework is illustrated in Figure 2. It consists of a modified CTR-GCN (Chen et al., 2021) based visual backbone and a Transformer (Vaswani et al., 2017) that takes in the visual features and generates the spoken translation.

Frame Pre-processing: To achieve end-to-end training on long video sequences, we choose to use pose keypoints extracted using MMPose (Contributors, 2020) as the input of our framework. This reduces the pressure on computing resources and enables us to use longer sequences of frames. Previous methods (Li et al., 2020b; Camgoz et al., 2020) mostly rely on pre-processed visual features extracted using models like I3D (Carreira and Zisserman, 2017) or CNN-based methods (Szegedy et al., 2017; Tan and Le, 2019). It has also been proved in this work (Camgoz et al., 2020) that a proper pre-training of the visual backbone can bring tremendous performance gain for the translation task. Then, it is a natural idea that we want to further improve the visual backbone through the supervision of the translation task. So we choose to use a lightweight GCN as our visual backbone and train the backbone all together.

Visual Backbone: The visual backbone takes in $T \times 76 \times 3$ keypoints including face, both hands and upper body. Each point contains 3 channels, which indicates the 2D position and a confidence value ranging from 0 to 1.0. The output feature of all the keypoints is pooled by regions at the end of the network and produces a 1024-dimensional feature. The multi-scale TCNs (Liu et al., 2020b) in the backbone downsample the temporal dimension by a factor of 4. The backbone is pre-trained on the WLASL dataset (Li et al., 2020a) through the isolated sign language recognition task.

Visual2Text Encoder: The visual2text encoder receives features from the visual backbone and translates these features from visual space to text space features $F_{enc} = \{s_1, s_2, \dots, s_N\}$. It provides context for the textual decoder and the encoded visual features are also passed to the contrastive concept mining module. The output visual features of the visual backbone are combined with a fixed sinusoidal position encoding following (Vaswani et al., 2017), which provides temporal

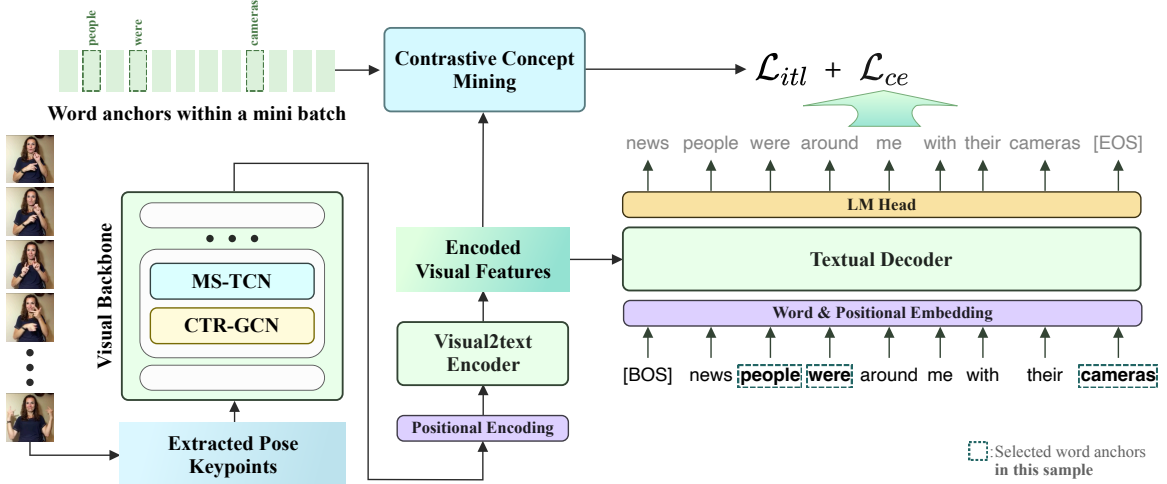


Figure 2: Overall illustration of the framework.

information for the encoder.

Textual Decoder: The textual decoder models the spoken translation in an auto-regressive manner. During the training phase, the spoken translation target Y is first tokenized using a BPE tokenizer (Sennrich et al., 2016) into $\hat{Y} = \hat{w}_{1:\hat{L}}$, which reduces out of vocabulary words during generation. Then we insert $\hat{w}_0 = [\text{BOS}]$ and $\hat{w}_{\hat{L}+1} = [\text{EOS}]$ at the start and end to indicate the beginning and end of the decoding process. The tokens \hat{Y} are converted into vectors through a word embedding layer and learned positional embedding, which is then summed together element-wise. Followed by layer normalization (Ba et al., 2016) and dropout (Srivastava et al., 2014). Then these vectors are passed through multiple transformer decoder layers to generate the feature $F_{dec} = \{r_0, r_1, \dots, r_{\hat{L}+1}\}$ for each token. The vectors are masked to ensure causality, one token can only interact with tokens that came beforehand. We share the learned word embedding weights with the language modeling head at the end of the decoder similar to (Press and Wolf, 2017; Desai and Johnson, 2021).

3.2 Cross-entropy Loss for Sign Translation

The language modeling head \mathcal{F}_{lm} in the textual decoder predicts probabilities over the token vocabulary.

$$p(x_i|x_{0:i-1}, F_{enc}) = \text{softmax}(\mathcal{F}_{lm}(r_{0:i-1})) \quad (1)$$

where x_i indicates the hypothesis's i_{th} token. Following previous literature on SLT, we use a cross-entropy loss at the training stage to supervise the

text generation process. We have:

$$\mathcal{L}_{ce} = - \sum_{i=0}^{\hat{L}} \log(p(x_i|x_{0:i-1}, F_{enc})) \quad (2)$$

This might be adequate for the translation of text pairs. Because for translating two text-based language inputs, the number of words for the text pair is similar (and there is no visual backbone too). But the number of frames of a sign video is much greater than either the number of corresponding glosses or spoken translation. It is very difficult for the encoder to learn a good representation as the token number of the encoder is much larger than that of the decoder, not to mention that we also want the encoder to provide good supervision for the visual backbone. In the work of Shi et al. (2022), they observed deteriorated performance if they tried to train the visual backbone and the transformer together. Thus we reckon in this case, a single cross-entropy loss at the end of the framework is not competent for our intended purpose.

3.3 Contrastive Concept Mining

Under the presumption that single cross-entropy loss is not enough. We want to provide additional supervision for the visual2text encoder. We intend to achieve such effect by exploiting the shared semantics between sign and text. A sign video can be roughly considered as multiple chunks (ignoring transition between signs), with each chunk of consecutive frames representing one sign word (a gloss). Though we cannot get the exact sign word for each chunk as the spoken translation does not

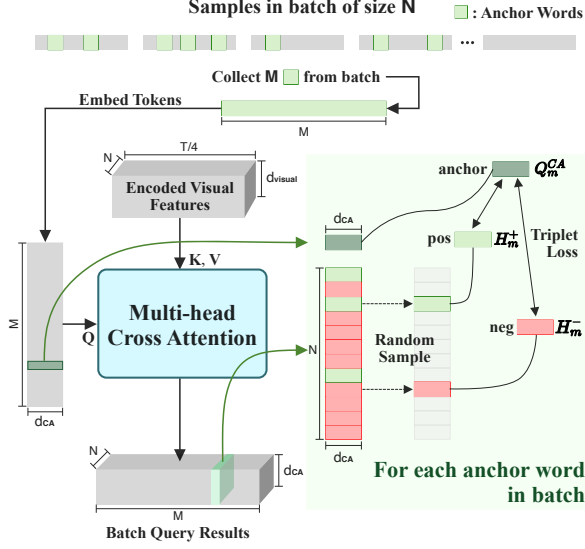


Figure 3: Overall illustration of the contrastive concept mining (CCM) process.

necessarily contains all the sign words and the orders also do not match. Key concepts expressed through sign and spoken translation should share the same underlying latent space. With this in mind, we design Contrastive Concept Mining (CCM) as shown in Figure 3.

The process of CCM consists of two steps: 1) Find possible words to be used as Conceptual Anchors (CA) in the training corpus, which we also refer to as anchor words. In practice, we mostly focus on verbs and nouns as we reckon such concepts are expressed in both the sign representation and the spoken language. It is natural to use these words as anchors for the encoder to structure the visual representations. 2) For each training batch of N samples, we collect all the anchor words (total of M words) in its spoken translations. For each word, we treat the sample containing such word as a positive sample and samples that do not as negative samples. Along with the global learned embedding for this word we conduct a triplet loss.

Global CA query on encoded feats: For a batch $B = \{x_1, x_2, \dots, x_N\}$ of N samples, we denote the collected word tokens as $B_{CA} = \{v_1, v_2, \dots, v_M\}$. M is the number of collected anchor words within the mini batch. These tokens are passed through an embedding layer to produce the query vector for multi-head cross attention.

$$Q^{CA} = \text{Embedding}_{CA}(B_{CA}) \quad (3)$$

where $Q^{CA} \in \mathbb{R}^{M \times d_{ca}}$, d_{ca} is the dimension

of the embedding layer for conceptual anchors. For output features of the encoder $F^{enc} = \{s_1, s_2, \dots, s_N\}$, in which $s_n \in \mathbb{R}^{L_{enc} \times d_{visual}}$. L_{enc} represents the max token length output by the encoder, d_{visual} is the dimension of the visual feature. The multi-head cross attention is defined as:

$$\begin{aligned} \text{CrossAtten}(Q^{CA}, s_n) &= [\text{head}_1 | \dots | \text{head}_h] W^o \\ \text{head}_i &= \text{Attention}(Q^{CA} W_i^Q, s_n W_i^K, s_n W_i^V) \end{aligned} \quad (4)$$

where $[\cdot]$ denotes the concatenation operation, head_i represents the output of the i -th head. The projection matrices are $W_i^Q \in \mathbb{R}^{d_{ca} \times d}$, $W_i^K, W_i^V \in \mathbb{R}^{d_{visual} \times d}$ and $W^o \in \mathbb{R}^{hd \times d_{ca}}$, in which d is the hidden dimension of the attention and d_{ca} is the final output dimension (same as the embedding dimension for CA). The attention process is defined as:

$$\text{Attention}(Q, K, V) = \text{softmax} \left(\frac{QK^T}{\sqrt{d}} \right) V \quad (5)$$

This process is repeated for each feature in F^{enc} . We denote $H_n = \text{CrossAtten}(Q^{CA}, s_n)$, we stack $\{H_n | n \in N\}$ to get the final output H of cross attention. We have $H_n \in \mathbb{R}^{M \times d_{ca}}$ and $H \in \mathbb{R}^{M \times N \times d_{ca}}$.

The cross-attention operation finds the most relevant part of an encoded visual feature to the CA query. The embedding of these word anchors Q^{CA} is shared across all the samples in the training set and is updated through the back-propagation process. We initialize these embeddings using pre-trained GloVe vectors (Pennington et al., 2014). The query results are the foundation for CCM, as we can encourage the encoder to gather visual information close to the word anchors and suppress noises similar to anchors but the anchor words are not presented in the sample.

Inter-sample triplet loss: We use a hinge-based triplet loss (Wang et al., 2014) as the learning objective for the query results H . The selection of positive and negative samples is carried out within a mini batch. For each unique CA v_m in a batch, we regard samples that contain this particular anchor word as positives and those that do not as negative samples. Since there might be more than one positive or negative sample for v_m , one positive or negative sample is chosen randomly. The objective function is formulated as:

$$\begin{aligned}
l_m &= \mu - \text{sim}(H_m^+, Q_m^{CA}) + \text{sim}(H_m^-, Q_m^{CA}) \\
\mathcal{L}_{itl} &= \max(0, \frac{1}{M} \sum_{m=1}^M l_m)
\end{aligned}
\tag{6}$$

where H_m^+ and H_m^- denotes the query results for the sampled positive and negative sample for v_m respectively. We use $\text{sim}(\cdot, \cdot)$ to calculate the cosine similarity as the distance between two vectors. μ is the margin for the triplet loss, it determines the gap between the distances of H_m^+ and H_m^- to the anchor Q_m^{CA} .

3.4 Training and Inference

Our framework is trained by the joint loss \mathcal{L} of cross-entropy loss \mathcal{L}_{ce} and conceptual contrastive loss \mathcal{L}_{itl} , which is formulated as:

$$\mathcal{L} = \mathcal{L}_{ce} + \lambda \mathcal{L}_{itl} \tag{7}$$

where λ is the hyper parameter that determines the scale of the inter-sample triplet loss. CCM only works during the training phase, and does not introduce additional parameters for inference.

4 Experiments

In this section we provide details on the datasets and translation protocol we follow. Along with quantitative and qualitative results on different benchmarks. We also give a deep analysis about the design components of our method.

4.1 Dataset and Protocols

OpenASL: OpenASL (Shi et al., 2022) is a large-scale American sign language dataset collected from online video sites. It covers a variety of domains with over 200 signers. With 98,417 translation pairs it’s the largest publicly available ASL translation dataset to date. 966 and 975 pairs are selected as validation and test sets respectively.

How2Sign: How2Sign (Duarte et al., 2021) is a large-scale American sign language dataset. It contains multi-modality data including video, speech, English transcript, keypoints, and depth. The signing videos are multi-view and performed by signers in front of a green screen. There are 31,128 training, 1,741 validation, and 2,322 test clips.

Gloss-free Sign2Text: *Sign2Text* directly translates from continuous sign videos to the corresponding spoken languages as proposed by Camgoz et al.

(2018). Unlike previous works, we ditch the need for gloss annotations throughout the entire framework including the pre-training phase.

Evaluation Metrics: To evaluate the translation quality, we report BLEU score (Papineni et al., 2002) and ROUGE-L F1-Score (Lin, 2004) following Camgoz et al. (2018). Same as OpenASL, we also report BLEURT score (Sellam et al., 2020). BLEURT is based on BERT (Devlin et al., 2019) and trained on rating data, it can correlate better with human judgments than BLEU and ROUGE.

4.2 Implementation Details

In our experiment, we use PyTorch (Paszke et al., 2019) to train the model on NVIDIA A100s. We rely on PyTorch’s implementation of Transformers to build the framework. We use byte pair encoding tokenizer provided by Huggingface’s Transformers (Wolf et al., 2020) library. The tokenizers are all trained from scratch on the training split of the corresponding datasets.

Network Details: We use multi-head attention with 4 heads in all transformer layers. The feed forward dimension in the transformer layers is set to 1024, and we use 4 layers both for encoders and decoders. For both OpenASL and How2Sign, we set the input frame cap to 512. The word embedding layer is trained from scratch with a dimension of 768.

Training & Testing: The model is trained using the AdamW (Loshchilov and Hutter, 2017) optimizer. We use a linear learning rate scheduler with 1000 warm-up steps. The learning rate is 3×10^{-4} with 400 epochs for both OpenASL and How2Sign. The models on OpenASL are trained across 4 GPUs with a batch size of 48 on each process for about 4 days. For How2Sign the model is trained across 8 GPUs with a batch size of 40 per process. In the text generation phase, we follow the common practice and use beam search with a beam size of 5.

Selection of anchor words: We rely on NLTK’s (Bird et al., 2009) default POS (part-of-speech) tagger to select words used in CCM. First, the training corpus is tokenized using NLTK’s punkt tokenizer. Then we pass the tokens to the POS tagger and filter out tags classified as general verbs or nouns (NN, NNP, NNS, VB, VBD, VBG, VBN, VBP, VBZ). Finally, we filter the verbs and nouns by their appearance frequency in the corpus. Words with occurrence not exceeding 10 or close to

the total sample count are discarded in this process.

4.3 Comparison with state-of-the-art

We test our framework on OpenASL against the multi-cue baseline proposed in the paper, as shown in Table 1. The baseline method incorporates multiple streams of global, mouth, and hands features and relies on external models to conduct sign spotting and fingerspelling sign search. Our framework, both GloFE-N (using only nouns as anchor words) and GloFE-VN (using both verbs and nouns as anchor words) surpasses all the previous methods on all metrics. The improvement on BLEURT stands out with a margin of 5.26 for GloFE-VN on the TEST set, which is 16.9% higher compared to the previous state-of-the-art. As for BLEURT on the DEV set, GloFE-N improves more than GloFE-VN with a gap of 6.08 over the previous state-of-the-art.

We obtain the best TEST result of 7.06 B4 with our VN model and the best DEV result of 7.51 B4 with the N model. Though the N model obtains significantly higher scores on the DEV set, results on the TEST set are lower than the VN model. The vocabulary size on N is close to VN (4, 238 to 5, 523), but as the N model only uses nouns the word type is less diverse. The lack of diversity makes the model less generalized, and more likely to fit the DEV set as it contains more similar samples to the training set.

We also test the framework on How2Sign. The results are shown in Table 2. We surpass the previous method on BLEU-4 but fall behind on the BLEU metric measuring smaller n-grams. The VN vocabulary size for How2Sign is around 2, 000 which is close to the number of test clips in How2Sign. Combined with the higher B4, it shows that our framework is better at generating short phrases. But the coverage of concepts is limited by the vocabulary size of the anchor words.

4.4 Ablation Study

4.4.1 Effect of Components

We examine the effectiveness of different design components as shown in Table 3. Namely, we ablate on the effect of the E2E (end-to-end training), PE (positional encoding for visual features), and CCM (contrastive concept mining), respectively. As a baseline, we first train a model without the three components. Without E2E, even we add PE and CCM both to the framework. The improvement

over baseline is only at 0.24 B4. If we add E2E back, this gap is widened significantly to 1.25 B4. This proves that our design can improve the visual backbone’s ability to recognize signs composed of multiple frames. With E2E, we also validate the effectiveness of PE and CCM, respectively. First, they both improve on the baseline line with a perceptible margin. When comparing PE to CCM, CCM is more performant, with an improvement of 0.65 B4 against 0.42 B4 over the baseline.

4.4.2 Type of Anchor Words

We study the type of words selected in this experiment. From Table 4 we can see that with V, N, and VN, model performance increase as the size of the vocabulary increases. But when we added A (adverbs and adjectives) to the vocab, the performance deteriorates by 0.43 B4. This is because the vocabulary jump from V to VN (or V to N), the number of conceptual word increases significantly. But with the addition of A, the extra words consists of major decorative purposes, they add to existing concepts (adverbs and adjectives modify verbs and nouns respectively). The number of conceptual word does not increase, but there are more anchors to attend to in the CCM process, which increases the learning difficulty.

4.4.3 Inter-sample Triplet Loss Weight

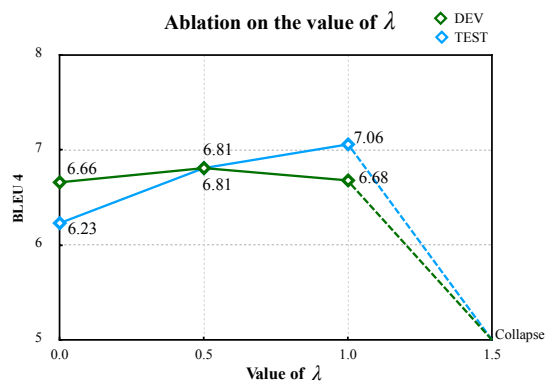


Figure 4: Ablation on the value of λ . Model collapsed when λ is at 1.5 or larger.

Here we study the effect of inter-sample triplet loss \mathcal{L}_{itl} by varying the weight λ . As shown in Figure 4, B4 on the DEV set fluctuates within a small range while B4 on the TEST set increased 0.83 as λ increases from 0 to 1.0. The model collapsed after λ goes beyond 1.5. When λ goes beyond 1.5, \mathcal{L}_{itl} takes the dominant spot in the combined loss.

Methods	DEV						TEST					
	ROUGE	BLEU-1	BLEU-2	BLEU-3	BLEU-4	BLEURT	ROUGE	BLEU-1	BLEU-2	BLEU-3	BLEU-4	BLEURT
Conv-GRU	16.25	16.72	8.95	6.31	4.82	25.36	16.10	16.11	8.85	6.18	4.58	25.65
I3D-transformer	18.88	18.26	10.26	7.17	5.60	29.17	18.64	18.31	10.15	7.19	5.66	28.82
OpenASL	20.43	20.10	11.81	8.43	6.57	31.22	21.02	20.92	12.08	8.59	6.72	31.09
GloFE-N (ours)	21.63	21.78	13.35	9.61	7.51	37.30	21.23	20.49	12.27	8.76	6.82	36.68
GloFE-VN (ours)	21.37	21.06	12.34	8.68	6.68	36.75	21.75	21.56	12.74	9.05	7.06	36.35

Table 1: Results on the OpenASL dataset. **N** represents the model trained using only nouns as anchor words and **VN** means the model is trained using both verbs and nouns as anchor words.

Methods	DEV						TEST					
	ROUGE	BLEU-1	BLEU-2	BLEU-3	BLEU-4	BLEURT	ROUGE	BLEU-1	BLEU-2	BLEU-3	BLEU-4	BLEURT
Alvarez†	-	17.73	7.94	4.13	2.24	-	-	17.40	7.69	3.97	2.21	-
GloFE-VN (ours)	12.98	15.21	7.38	4.07	2.37	30.95	12.61	14.94	7.27	3.93	2.24	31.65

Table 2: Results on the How2Sign dataset. †: Alvarez et al. used CNN model (Koller et al., 2019) pre-trained with GLOSS annotations to extract the visual features.

E2E	PE	CCM	RG	B@1	B@2	B@3	B@4	BLEURT
			18.94	18.25	10.38	7.37	5.81	34.35
	✓	✓	20.24	19.20	11.10	7.82	6.05	35.39
✓	✓		20.92	20.37	11.62	8.09	6.23	35.65
✓		✓	20.48	19.71	11.48	8.20	6.46	35.96
✓	✓	✓	21.75	21.56	12.74	9.05	7.06	36.35

Table 3: Ablation on OpenASL demonstrates the effect of our different components. **E2E**: Whether to conduct end-to-end training to train the visual backbone together. **PE**: Fixed sinusoidal positional encoding added to the input of visual2text encoder. **CCM**: Whether to use Contrastive Concept Mining on the encoded visual features during the training phase. B@N represents the BLEU-N score, this also applies to tables that came after this.

Word Type	Vocab.	RG	B@1	B@2	B@3	B@4	BLEURT
V	1693	20.96	20.70	11.89	8.42	6.51	36.50
N	4238	21.23	20.49	12.27	8.76	6.82	36.68
VN	5523	21.75	21.56	12.74	9.05	7.06	36.35
VNA	6726	20.85	20.22	11.88	8.44	6.63	35.90

Table 4: Ablation on selecting different types of words as the conceptual anchors. **V** and **N** stands for verb and noun respectively. **A** stands for adverbs and adjectives. The sum of the vocab size of **V** and **N** individually is greater than **VN** because the word type can vary depending on its relative position in the sentence.

But \mathcal{L}_{itl} alone cannot guide the generation process, resulting in the collapse of the model.

4.5 Qualitative Results

Table 5 shows the qualitative results of the generated text of GloFE compared to the baseline model. We mainly focus on whether the model generates the same conceptual words (verbs and

Ref:	today is the first day of winter .
Baseline:	today is the first day of the day .
GloFE:	today is the first day of the winter day.
Ref:	meteorologists say freeze warnings remain in the south including florida.
Baseline:	officials are warning about 200 feet of snow.
GloFE:	meteorologists say the weather will be keeping in louisiana.
Ref:	we have also reached out to ntid and asked for their response .
Baseline:	we also reached out to the nad board members for their stories .
GloFE:	we also reached out to you for their response .
Ref:	the death toll from hurricane dorian is rising in the bahamas.
Baseline:	the death toll is now emotional ."
GloFE:	and the death toll in the bahamas is rising .

Table 5: Qualitative results of GloFE on the TEST split of OpenASL compared to the baseline model first introduced in 4.4.1. We use **red** to indicate mistranslation of conceptual words and **green** to show the matching concepts between GloFE and Ref.

nouns) as the reference text. For each sample, we show the reference text and the text generated by the baseline model and GloFE. We use red to indicate the mistranslated conceptual words in the baseline results and green to show the matching concepts. In the first example, both baseline and GloFE generate similar text with one key difference. GloFE successfully captures the concept of winter (noun) in the sign expression while the baseline does not. However, GloFE cannot always capture all the correct concepts. In the third example, GloFE failed the capture ntid and asked. But compared to the baseline, GloFE still managed to

translate response correctly. In general, GloFE is capable of generating a more accurate translation of objects and motions expressed in the signing sequence.

5 Conclusion

In this paper, we propose a novel gloss-free end-to-end framework for sign language translation. Design an intermediate representation that can act as a fill-in when gloss annotation is not available. We exploit the shared semantics between sign and text, by extracting common conceptual words from the spoken translation. The model is trained end-to-end including the visual backbone, no gloss is used in training or pre-training, and achieves state-of-the-art performance on the largest sign languages translation dataset publicly available.

Limitations

Our model is trained in an end-to-end manner, resulting in more training time costs than feature-based methods. To eliminate the need for gloss annotations, the CCM process relies on a large amount of sign and translation pairs. The generalizability of the model is restrained by the number of such pairs available. The more ideal end-to-end framework should combine the visual backbone and visual2text encoder into one visual encoder that can be trained end-to-end. In addition, the selection of conceptual words is done according to manually-designed rules now and relies on external toolkits like NLTK. We will investigate automatic conceptual word extraction methods in future work.

Ethics Statement

Our work focuses on the task of sign language translation. Such systems aims to use technology to facilitate the day-to-day life of the deaf and hard-of-hearing community. Though we improve on the baseline, the proposed model still does not equip with the ability to serve as an interpreter in real-life scenarios. We use extracted keypoints as the input of the model, there are little to no concerns about personal privacy. For now, the model is only validated on American sign language datasets, currently it's not able to help people that do not use ASL.

Acknowledgements

This work is supported by the Fundamental Research Funds for the Central Universities (No. 226-

2022-00051), and also supported in part by the Natural Science Foundation of Zhejiang Province (DT23F020008).

References

- Aishwarya Agrawal, Damien Teney, and Aida Nematzadeh. 2022. **Vision-language pretraining: Current trends and the future**. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*, pages 38–43, Dublin, Ireland. Association for Computational Linguistics.
- Patricia Cabot Alvarez, Xavier Giró Nieto, and Laia Tarés Benet. Sign language translation based on transformers for the how2sign dataset.
- Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450*.
- Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc."
- Necati Cihan Camgoz, Simon Hadfield, Oscar Koller, Hermann Ney, and Richard Bowden. 2018. Neural sign language translation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7784–7793.
- Necati Cihan Camgoz, Oscar Koller, Simon Hadfield, and Richard Bowden. 2020. Sign language transformers: Joint end-to-end sign language recognition and translation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10023–10033.
- Joao Carreira and Andrew Zisserman. 2017. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308.
- Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. 2020. Uniter: Universal image-text representation learning. In *ECCV*.
- Yutong Chen, Fangyun Wei, Xiao Sun, Zhirong Wu, and Stephen Lin. 2022. A simple multi-modality transfer learning baseline for sign language translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5120–5130.
- Yuxin Chen, Ziqi Zhang, Chunfeng Yuan, Bing Li, Ying Deng, and Weiming Hu. 2021. Channel-wise topology refinement graph convolution for skeleton-based action recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13359–13368.

- MMPose Contributors. 2020. Openmmlab pose estimation toolbox and benchmark. <https://github.com/open-mmlab/mmpose>.
- Karan Desai and Justin Johnson. 2021. Virtex: Learning visual representations from textual annotations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11162–11173.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Amanda Duarte, Shruti Palaskar, Lucas Ventura, Deepti Ghadiyaram, Kenneth DeHaan, Florian Metze, Jordi Torres, and Xavier Giro-i Nieto. 2021. How2sign: A large-scale multimodal dataset for continuous american sign language. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2735–2744.
- Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. 2006. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*, pages 369–376.
- Tao Jin, Zhou Zhao, Meng Zhang, and Xingshan Zeng. 2022. Prior knowledge and memory enriched transformer for sign language translation. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 3766–3775, Dublin, Ireland. Association for Computational Linguistics.
- Youngmin Kim, Minji Kwak, Dain Lee, Yeongeun Kim, and Hyeongbo Baek. 2022. Keypoint based sign language translation without glosses. *arXiv preprint arXiv:2204.10511*.
- Sang-Ki Ko, Chang Jo Kim, Hyedong Jung, and Choongsang Cho. 2019. Neural sign language translation based on human keypoint estimation. *Applied sciences*, 9(13):2683.
- Oscar Koller, Necati Cihan Camgoz, Hermann Ney, and Richard Bowden. 2019. Weakly supervised learning with multi-stream cnn-lstm-hmms to discover sequential parallelism in sign language videos. *IEEE transactions on pattern analysis and machine intelligence*, 42(9):2306–2320.
- Dongxu Li, Cristian Rodriguez, Xin Yu, and Hongdong Li. 2020a. Word-level deep sign language recognition from video: A new large-scale dataset and methods comparison. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 1459–1469.
- Dongxu Li, Chenchen Xu, Xin Yu, Kaihao Zhang, Benjamin Swift, Hanna Suominen, and Hongdong Li. 2020b. Tspnet: Hierarchical feature learning via temporal semantic pyramid for sign language translation. *Advances in Neural Information Processing Systems*, 33:12034–12045.
- Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.
- Yan Ling, Jianfei Yu, and Rui Xia. 2022. Vision-language pre-training for multimodal aspect-based sentiment analysis. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2149–2159, Dublin, Ireland. Association for Computational Linguistics.
- Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020a. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:726–742.
- Ziyu Liu, Hongwen Zhang, Zhenghao Chen, Zhiyong Wang, and Wanli Ouyang. 2020b. Disentangling and unifying graph convolutions for skeleton-based action recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 143–152.
- Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.
- Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.
- Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.
- Ofir Press and Lior Wolf. 2017. Using the output embedding to improve language models. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 157–163, Valencia, Spain. Association for Computational Linguistics.

- Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *ICML*.
- Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.
- Ryokan Ri and Yoshimasa Tsuruoka. 2022. [Pretraining with artificial language: Studying transferable knowledge in language models](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7302–7315, Dublin, Ireland. Association for Computational Linguistics.
- Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. [BLEURT: Learning robust metrics for text generation](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892, Online. Association for Computational Linguistics.
- Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. [Neural machine translation of rare words with subword units](#). In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.
- Paul Hongsuck Seo, Arsha Nagrani, Anurag Arnab, and Cordelia Schmid. 2022. End-to-end generative pre-training for multimodal video captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17959–17968.
- Bowen Shi, Diane Brentari, Greg Shakhnarovich, and Karen Livescu. 2022. Open-domain sign language translation learned from online video. In *EMNLP*.
- Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958.
- Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. 2017. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Thirty-first AAAI conference on artificial intelligence*.
- Hao Tan and Mohit Bansal. 2019. Lxmert: Learning cross-modality encoder representations from transformers. In *EMNLP*.
- Mingxing Tan and Quoc Le. 2019. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.
- Jiang Wang, Yang Song, Thomas Leung, Chuck Rosenberg, Jingbin Wang, James Philbin, Bo Chen, and Ying Wu. 2014. Learning fine-grained image similarity with deep ranking. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1386–1393.
- Zirui Wang, Jiahui Yu, Adams Wei Yu, Zihang Dai, Yulia Tsvetkov, and Yuan Cao. 2021. Simvlm: Simple visual language model pretraining with weak supervision. In *ICLR*.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. [Transformers: State-of-the-art natural language processing](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.
- Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. 2022. Coca: Contrastive captioners are image-text foundation models. *arXiv preprint arXiv:2205.01917*.
- Hao Zhou, Wengang Zhou, Weizhen Qi, Junfu Pu, and Houqiang Li. 2021. Improving sign language translation with monolingual data by sign back-translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1316–1325.

ACL 2023 Responsible NLP Checklist

A For every submission:

- A1. Did you describe the limitations of your work?
Dedicated section: Limitations
- A2. Did you discuss any potential risks of your work?
Ethic statements
- A3. Do the abstract and introduction summarize the paper's main claims?
abstract and section 1
- A4. Have you used AI writing assistants when working on this paper?
Left blank.

B Did you use or create scientific artifacts?

section 3

- B1. Did you cite the creators of artifacts you used?
section 3 and 4.1
- B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Not applicable. Left blank.
- B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Not applicable. Left blank.
- B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Not applicable. Left blank.
- B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
section 4.1
- B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
section 4.1

C Did you run computational experiments?

section 4.3 4.4

- C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
section 4.2

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?

section 4.2

C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?

section 4.3

C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?

section 4.2

D Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?

Not applicable. Left blank.

D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?

Not applicable. Left blank.

D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?

Not applicable. Left blank.

D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?

Not applicable. Left blank.

D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?

Not applicable. Left blank.