Exploring Phoneme-Level Speech Representations for End-to-End Speech Translation

Elizabeth Salesky[1], **Matthias Sperber**[2], **Alan W Black**[1]
[1]Carnegie Mellon University, USA
[2]Karlsruhe Institute of Technology, Germany
esalesky@cs.cmu.edu

Abstract

Previous work on end-to-end translation from speech has primarily used frame-level features as speech representations, which creates longer, sparser sequences than text. We show that a naïve method to create compressed phoneme-like speech representations is far more effective and efficient for translation than traditional frame-level speech features. Specifically, we generate phoneme labels for speech frames and average consecutive frames with the same label to create shorter, higher-level source sequences for translation. We see improvements of up to 5 BLEU on both our high and low resource language pairs, with a reduction in training time of 60%. Our improvements hold across multiple data sizes and two language pairs.

1 Introduction

The way translation input is represented has been shown to impact performance as well as how much data the model requires to train (Sennrich et al., 2016; Salesky et al., 2018; Cherry et al., 2018). The current standard approach for text-based translation is to segment words into subword units as a preprocessing step (Sennrich et al., 2016). Clustering common character sequences increases frequency of units in data and improves generalization to new word forms and rare words.

End-to-end speech-to-text models are showing competitive results (Weiss et al., 2017; Bansal et al., 2018a,b; Bérard et al., 2018; Anastasopoulos and Chiang, 2018), but so far have not compared different ways to represent speech input. Unlike text, where discrete trainable embeddings are typically used, speech models typically use continuous features extracted from sliding windows (frames), held fixed during training. Frame-level features yield significantly longer, more sparsely-represented sequences than their text equivalents, and so speech models stand to benefit from learning compressed input representations. Previous works have reduced sequence lengths to make training more tractable through fixed-length downsampling. However, phonemes are variable lengths. Other work has shown promising results using phonemic representations and unsupervised term discovery from variable length sequences in MT and other domains, but as discrete units (Wilkinson et al., 2016; Bansal et al., 2017; Adams et al., 2016; Kamper et al., 2016; Dalmia et al., 2018b; Chung and Glass, 2018). Inspired by these works, we explore higher-level continuous speech embeddings for end-to-end speech translation.

Specifically, we use alignment methods to generate phoneme labels, and average consecutive frames with the same label to create phoneme-like feature vectors from variable numbers of frames. We use the Fisher Spanish-English and low-resource Mboshi-French datasets. We compare performance on the full Fisher dataset to smaller subsets as in Bansal et al. (2018b). As it is not possible to train a high-performing recognizer on many lower-resource tasks, we use a high-resource model applied cross-lingually to create phoneme labels for Mboshi. We show significant performance improvements and reductions in training time under all conditions, demonstrating phoneme-informed speech representations are an effective and efficient tool for speech translation.

2 Method

While frame-level Mel-frequency cepstral coefficient (MFCC) and filterbank features are informative, they create long, repetitive sequences which take recurrent models many examples to learn to model. Higher-level representations like phonemes can create shorter, better-represented input sequences to improve training efficiency and

Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 1835–1841
Florence, Italy, July 28 - August 2, 2019.

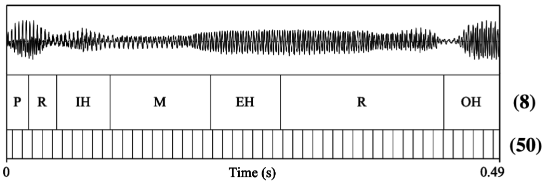

Figure 1: Example comparing number of frame-level features (50) to phoneme alignments (8). We saw an average reduction in sequence length of ∼80%.

model robustness. Here, we average frame-level features within phoneme-like units to create one representation from a variable number of frames, using a trained speech recognizer and alignment.

We extract 40-dimensional Mel filterbank features with per-speaker mean and variance normalization using Kaldi (Povey et al., 2011). Using an HMM/DNN system trained on the full Fisher Spanish dataset using the Kaldi (Povey et al., 2011) recipe for Fisher Spanish, we compute phoneme alignments using the triphone model (tri3a). 50 phoneme labels are used, including variants of silence, noise, and laughter. Within each utterance, we average the feature vectors for consecutive frames with the same label.

The above method requires a recognizer with reasonable performance to perform alignment, not possible in low-resource conditions. Therefore, for Mboshi, we use a method that does not require language-specific data to generate a phoneme-like sequence. Specifically, we apply a Connectionist Temporal Classification (CTC) model trained with 6000 hours of English data (notably, not a related language), as described in Dalmia et al. (2018b) with the features from Dalmia et al. (2018a). To train this model, three frame-level features are spliced together, so output labels apply to a span of three frames. Labels comprise a set of 40 phonemes and the CTC 'blank' where the model is uncertain. The CTC 'blank' transition label enables all frames to be aligned to a label. As above, we average the feature vectors for consecutive frames with the same label within an utterance.

3 Model Architecture

As in Bansal et al. (2018a), we use a sequence-to-sequence architecture inspired by Weiss et al. but modified to train within available resources; specifically, all models may be trained in less than 5 days on one GPU. We build an encoder-decoder model with attention in `xnmt` (Neubig et al., 2018) with 512 hidden units throughout. We use a 3-layer BiLSTM encoder. We do not use the additional convolutional layers from Weiss et al. and Bansal et al. to reduce temporal resolution, but rather use network-in-network (NiN) projections from previous work in sequence-to-sequence ASR (Zhang et al., 2017; Sperber et al., 2018) to get the same total 4× downsampling in time. This gives the benefit of added depth with fewer parameters. We compare our performance to these two works in Section 5.1. We closely follow the LSTM/NiN encoder used in Sperber et al. (2018) for ASR and use the same training procedure, detailed in Appendix A. We use an MLP attention with 1 hidden layer with 128 units and 64-dimensional target embeddings, though we use only 1 decoder hidden layer as opposed to 3 or 4 in previous works. All models use the same target preprocessing as previous work on this dataset: lowercasing and removing punctuation aside from apostrophes.

4 Datasets

Spanish-English. We use the Fisher Spanish speech corpus (Graff et al.), which consists of 160 hours of telephone speech in multiple Spanish dialects split into 138K utterances, translated via crowdsourcing by Post et al. (2013). We use the standard dev and test sets, each with ∼4k utterances. We do not use dev2. Four reference translations are used to score dev and test.

Mboshi-French. Mboshi is a Bantu language spoken in the Republic of Congo with ∼160k speakers. We use the Mboshi-French parallel corpus (Godard et al., 2017) for our low-resource setting, which contains <5 hours of speech split into training and development sets of 4616 and 500 utterances respectively. This corpus does not have a designated test set, so as in Bansal et al. (2018b) we removed 200 randomly sampled utterances from training for development data and use the designated development set as test.

5 Results

5.1 Baseline

We first compare our model to previously reported end-to-end neural speech translation results on the Fisher Spanish-English task using frame-level features. Table 1 shows our results on the full training set with comparisons to Weiss et al. (2017) and Bansal et al. (2018a). Weiss et al.'s model is

	Weiss et al.		Bansal et al.		Ours	
	dev	**test**	**dev**	**test**	**dev**	**test**
BLEU	46.5	47.3	29.5	29.4	32.4	33.7

Table 1: Single task end-to-end speech translation BLEU scores on full dataset.

significantly deeper than ours, with 4 more encoder layers and 3 more decoder layers. After more than two weeks of expensive multi-GPU training, it reaches a 4-reference BLEU score of 47.3 on `test`. We, like Bansal et al. (2018a,b), made modifications to our architecture and training schemes to train on a single GPU in approximately five days. While Bansal et al. use words on the target side to reduce time to convergence at a slight performance cost, we are able to use characters as in Weiss et al. by having a still shallower architecture (2 fewer layers on both the encoder and decoder), which allows us to translate to characters with approximately the same training time per epoch they observe with words ($\sim$2 hours). We converge to a four-reference test BLEU of 33.7, showing 3-4 BLEU improvements over Bansal et al. (2018a) on dev and test. This demonstrates that our model has reasonable performance, providing a strong baseline before turning to our targeted task comparing input representations.

5.2 Frames vs Phonemes

On our target task, we compare different subsets of the data to see how our method compares under different data conditions, using the full 160 hours as well as 40 and 20 hour subsets. Table 2 shows our results using frame vs phoneme-level speech input. When we use our phoneme-like embeddings, we see relative performance improvements of 13% on all data sizes, or up to 5.2 BLEU on the full dataset. Further, in reducing source lengths by $\sim$80%, training time is improved. We saw an average reduction in training time of 61%, which for the full dataset means we were able to train our model in 39.5 hours rather than 118.2.

	Frames		Phonemes		BLEU	Time
Data	**dev**	**test**	**dev**	**test**	Δ	Δ
Full	32.4	33.7	37.6	38.8	+5.2	–67%
40hr	19.5	17.4	21.0	19.8	+2.0	–52%
20hr	9.8	8.9	11.1	10.0	+1.2	–65%

Table 2: Comparison of frame vs phoneme input on Spanish-English SLT, with average BLEU improvement and average reduction in training time.

We compare our variable-length downsampling to fixed-stride downsampling by striding input frames. With a fixed stride of 2, performance decreases on 40 hours by $\sim$2 BLEU from 19.5 to 17.0 on dev and 17.4 to 15.6 on test. With a fixed stride of 3, performance drops further to 13.7 and 11.8, respectively. By contrast, we saw improvements of +2 BLEU on 40 hours using our variable-length downsampling, though it lead to greater reductions in the number of input feature vectors. Clearly phoneme-informed reduction is far more effective than fixed schedule downsampling.

5.3 Analysis

To better understand our improvements, we target three points. Does making source and target sequence lengths more well-matched improve performance? To test we compare target preprocessing granularities. Second, reducing source lengths will impact both the encoder and attention. To investigate, we look at both encoder downsampling and ASR, where unlike MT, sequences are monotonic. Finally, we look at our low-resource case, Mboshi-French, where we must get phoneme labels from a cross-lingual source.

Previous work on sequence-to-sequence speech translation has used encoder downsampling of $4\times$, while $8\times$ is more common among sequence-to-sequence ASR systems (Zhang et al., 2017), motivated by reducing parameters and creating more one-to-one relationships between lengths of target sequence (typically characters) and the final encoder states to which the model attends. We use encoder downsampling of $4\times$, concatenating adjacent states after each layer. Table 3 shows target sequence lengths and results with different preprocessing. By averaging frames per local phoneme label in addition to encoder downsampling, source sequence lengths are further reduced on average by 79%, yielding final encoder state lengths of 22, closest in length to 1k BPE targets (14) rather than characters (50). Given that the 1k BPE model per-

Target	Target	Frames		Phonemes	
Preproc.	Length	dev	test	dev	test
chars	50.2	18.8	**17.3**	20.0	18.4
1k bpe	13.7	**19.5**	**17.4**	**21.0**	**19.8**
10k bpe	10.6	16.2	14.7	18.4	17.5
words	10.4	16.4	14.6	18.2	17.4

Table 3: Comparing effects of target preprocessing with different sources on BLEU, Spanish-English 40hr

forms best, it does appear that more similar source and target lengths boost performance.

For Spanish, we found that the mean number of frames per phone was 7.6, while the median was 6. Silence in this dataset skews these statistics higher; silence-marked frames account for 10.7% of phone occurrences. Reducing multiple frames per phone to a single feature vector allows faster parameter optimization, as shown by improvements in early epochs in Figure 2.

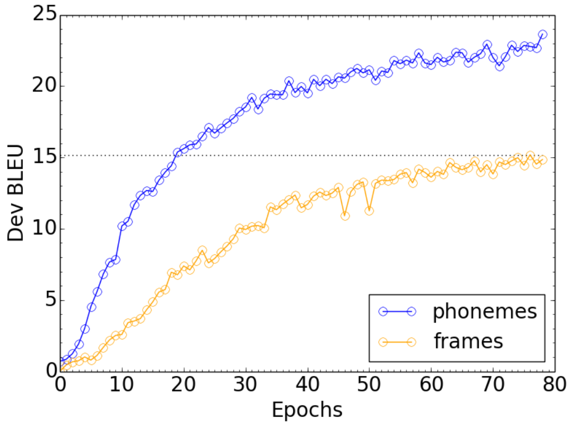

Figure 2: Dev BLEU over training with frames vs phonemes. Single-reference BLEU on 1k lines of dev.

We also compare the best phoneme models without encoder downsampling; with reduced sequence lengths, this becomes more tractable to train. On the full data, we see this improves our scores slightly, from 37.6 to 38.1 on dev and 38.8 to 39.2 on test. We see further improvements on 40 hours (22.4 dev & 20.3 test), and on 20 hours, similar dev performance but slight improvements on test (10.3 dev & 9.6 test). It is possible that with less data, the additional encoder parameters without downsampling do not receive enough updates to be well-trained.

To test whether the approach is a generally more effective input representation or only an aid in the particularly complex learning task of speech translation where it helps to reduce the distance between inputs and outputs, we apply our method to ASR, where are alignments are monotonic. We see similar levels of improvement, suggesting this approach produces generally more effective input representations: ∼18% relative improvements on all three dataset sizes, or up to –9 absolute WER on 40 and 20 hours, as detailed in Table 4. We note that Weiss et al. (2017) reported 25.7, 23.2 on dev and test, respectively, with a considerably larger network, which we are now able to match on test. We note that this neural model also outperforms the Kaldi models; the Kaldi model using the tri3a alignments we use for phoneme boundaries yields 45.7 dev WER, and using more sophisticated alignment models, achieves 29.8.

	Frames		Phonemes		WER	Time
Data	**dev**	**test**	**dev**	**test**	Δ	Δ
Full	33.4	30.0	28.0	23.4	–6.0	–43%
40hr	44.8	46.7	36.6	36.6	–9.2	–40%
20hr	56.3	59.1	48.2	49.1	–9.1	–50%

Table 4: Comparison of frame vs phoneme input on Spanish ASR, with average reduction in WER and average reduction in training time.

On our low-resource Mboshi task, we do not have enough data to train a high-quality recognizer to produce phoneme alignments. Instead, we use a model from an unrelated language (English) applied cross-lingually. With small training and evaluation sets, scores are less stable and changes must be taken with a grain of salt. We see very low scores with frames, but still see improvements with phonemes, though the labels were produced by an English model. Bansal et al. (2018b) reported 3.5 BLEU using frames, which they improved to 7.1 by pretraining their encoder with 300 hours of English and decoder with 20 hours of French. Creating phoneme-level embeddings, we are able to get similar levels of improvement without training the network on more data, though we use an unadapted foreign language model.

	Frames		Phonemes	
Data	**dev**	**test**	**dev**	**test**
Mboshi (chars)	0.0	0.0	5.2	3.6
Mboshi (1k bpe)	2.3	1.4	7.0	5.6
Mboshi (words)	1.8	1.4	7.8	5.9

Table 5: Comparison of frame vs phoneme input on Mboshi-French SLT. Mboshi phoneme labels produced with English CTC phoneme recognizer.

While LSTM-based sequence-to-sequence models are able to learn from long, redundant sequences, we show that they learn more efficiently and effectively across multiple data conditions when given sequences reduced using phoneme boundaries. This is evidenced by our improvements across all data sizes, and significant improvements in early epochs, shown in Figure 2.

We compared two methods for alignment, an HMM-based model and a CTC-based model, the first applied monolingually and the second cross-lingually. The CTC model yields blank alignments for some frames, reducing the range of frames to be averaged, though the center of mass often remains the same. We hypothesize that this does not greatly impact results, and previous work has explored using the middle HMM state for alignments rather than all (Stuker et al., 2003), but this would benefit from a more robust comparison. As well, a deeper comparison of monolingual versus cross-lingual alignments applied to a greater number of test languages would be beneficial.

6 Conclusion

Previous work on end-to-end speech translation has used frame-level speech features. We have shown that a naïve method to create higher-level speech representations for translation can be more effective and efficient than traditional frame-level features. We compared two input representations for two unrelated languages pairs, and a variety of differently-resourced conditions, using both a supervised alignment method and a cross-lingual method for our low-resource case. Our method does not introduce additional parameters: we hope to motivate future work on learning speech representations, with continued performance on lower-resource settings if additional parameters are introduced.

Acknowledgements We would like to thank the anonymous reviewers for their helpful comments.

References

Oliver Adams, Graham Neubig, Trevor Cohn, Steven Bird, Quoc Truong Do, and Satoshi Nakamura. 2016. Learning a lexicon and translation model from phoneme lattices. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2377–2382.

Antonios Anastasopoulos and David Chiang. 2018. Tied multitask learning for neural speech translation. *arXiv preprint arXiv:1802.06655*.

Sameer Bansal, Herman Kamper, Karen Livescu, Adam Lopez, and Sharon Goldwater. 2018a. Low-resource speech-to-text translation. *Proc. of Interspeech*. ArXiv:1803.09164.

Sameer Bansal, Herman Kamper, Karen Livescu, Adam Lopez, and Sharon Goldwater. 2018b. Pre-training on high-resource speech recognition improves low-resource speech-to-text translation. *arXiv preprint arXiv:1809.01431*.

Sameer Bansal, Herman Kamper, Adam Lopez, and Sharon Goldwater. 2017. Towards speech-to-text translation without speech recognition. *arXiv preprint arXiv:1702.03856*.

Alexandre Bérard, Laurent Besacier, Ali Can Kocabiyikoglu, and Olivier Pietquin. 2018. End-to-end automatic speech translation of audiobooks. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6224–6228. IEEE.

William Chan, Navdeep Jaitly, Quoc Le, and Oriol Vinyals. 2016. Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. In *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*, pages 4960–4964. IEEE.

Colin Cherry, George Foster, Ankur Bapna, Orhan Firat, and Wolfgang Macherey. 2018. Revisiting character-based neural machine translation with capacity and compression. *arXiv:1808.09943*.

Yu-An Chung and James Glass. 2018. Speech2vec: A sequence-to-sequence framework for learning word embeddings from speech. *arXiv preprint arXiv:1803.08976*.

Siddharth Dalmia, Xinjian Li, Florian Metze, and Alan W. Black. 2018a. Domain robust feature extraction for rapid low resource asr development. *2018 IEEE Workshop on Spoken Language Technology (SLT)*.

Siddharth Dalmia, Ramon Sanabria, Florian Metze, and Alan W. Black. 2018b. Sequence-based multi-lingual low resource speech recognition. *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4909–4913.

Yarin Gal and Zoubin Ghahramani. 2016. A theoretically grounded application of dropout in recurrent neural networks.

Pierre Godard, Gilles Adda, Martine Adda-Decker, Juan Benjumea, Laurent Besacier, Jamison Cooper-Leavitt, Guy-Noël Kouarata, Lori Lamel, Hélène Maynard, Markus Müller, et al. 2017. A very low resource language speech corpus for computational language documentation experiments. *arXiv preprint arXiv:1710.03501*.

David Graff, Shudong Huang, Ingrid Cartagena, Kevin Walker, and Christopher Cieri. Fisher spanish speech (LDC2010S01). Https://catalog.ldc.upenn.edu/ ldc2010s01.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Comput.*

Herman Kamper, Aren Jansen, and Sharon Goldwater. 2016. Unsupervised word segmentation and lexicon discovery using acoustic word embeddings. *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, 24(4):669–679.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. *Proc. of ICLR*. ArXiv:1412.6980.

Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation.

Graham Neubig, Matthias Sperber, Xinyi Wang, Matthieu Felix, Austin Matthews, Sarguna Padmanabhan, Ye Qi, Devendra Singh Sachan, Philip Arthur, Pierre Godard, et al. 2018. Xnmt: The extensible neural machine translation toolkit. *arXiv:1803.00188*.

Toan Q Nguyen and David Chiang. 2018. Improving lexical choice in neural machine translation. *Proc. of NAACL HLT*. ArXiv:1710.01329.

Matt Post, Gaurav Kumar, Adam Lopez, Damianos Karakos, Chris Callison-Burch, and Sanjeev Khudanpur. 2013. Improved speech-to-text translation with the fisher and callhome spanish–english speech translation corpus.

Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, et al. 2011. The kaldi speech recognition toolkit.

Elizabeth Salesky, Andrew Runge, Alex Coda, Jan Niehues, and Graham Neubig. 2018. Optimizing segmentation granularity for neural machine translation. *arXiv:1810.08641*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *ACL*.

Matthias Sperber, Jan Niehues, Graham Neubig, Sebastian Stüker, and Alex Waibel. 2018. Self-attentional acoustic models. *Proc. of EMNLP*. ArXiv:1803.09519.

Sebastian Stuker, Tanja Schultz, Florian Metze, and Alex Waibel. 2003. Multilingual articulatory features. In *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03).*, volume 1, pages I–I. IEEE.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision.

Ron J Weiss, Jan Chorowski, Navdeep Jaitly, Yonghui Wu, and Zhifeng Chen. 2017. Sequence-to-sequence models can directly transcribe foreign speech. *arXiv:1703.08581*.

Andrew Wilkinson, Tiancheng Zhao, and Alan W Black. 2016. Deriving phonetic transcriptions and discovering word segmentations for speech-to-speech translation in low-resource settings. In *INTERSPEECH*.

Yu Zhang, William Chan, and Navdeep Jaitly. 2017. Very deep convolutional networks for end-to-end speech recognition.

A Appendix. LSTM/NiN Encoder and Training Procedure Details

A.1 Encoder Downsampling Procedure

Weiss et al. (2017) and Bansal et al. (2018a) use two strided convolutional layers atop three bidirectional long short-term memory (LSTM) (Hochreiter and Schmidhuber, 1997) layers to downsample input sequences in **time** by a total factor of 4. Weiss et al. (2017) additionally downsample **feature** dimensionality by a factor of 3 using a ConvLSTM layer between their convolutional and LSTM layers. This is in contrast to the pyramidal encoder (Chan et al., 2016) from sequence-to-sequence speech recognition, where pairs of consecutive layer outputs are concatenated before being fed to the next layer to halve the number of states between layers.

To downsample in time we instead use the LSTM/NiN model used in Sperber et al. (2018) and Zhang et al. (2017), which stacks blocks consisting of an LSTM, a network-in-network (NiN) projection, layer batch normalization and then a ReLU non-linearity. NiN denotes a simple linear projection applied at every timestep, performing downsampling by a factor of 2 by concatenating pairs of adjacent projection inputs. The LSTM/NiN blocks are extended by a final LSTM layer for a total of three BiLSTM layers with the same total downsampling of 4 as Weiss et al. (2017) and Bansal et al. (2018a). These blocks give us the benefit of added depth with fewer parameters.

A.2 Training Procedure

We follow the training procedure from Sperber et al. (2018). The model uses variational recurrent dropout with probability 0.2 and target character dropout with probability 0.1 (Gal and Ghahramani, 2016). We apply label smoothing (Szegedy et al., 2016) and fix the target embedding norm to 1 (Nguyen and Chiang, 2018). For inference, we use a beam size of 15 and length normalization with exponent 1.5. We set the batch size dynamically depending on the input sequence length such that the average batch size was 36. We use Adam (Kingma and Ba, 2015) with initial learning rate of 0.0003, decayed by 0.5 when validation BLEU did not improve over 10 epochs initially and 5 epochs after the first decay. We do not use L2 weight decay or Gaussian noise, and use a single model replica. All models use the same preprocessing as previous work on this dataset: lowercasing and removing punctuation aside from apostrophes. We use input feeding (Luong et al., 2015), and we exclude utterances longer than 1500 frames to manage memory requirements.