MeLT: Message-Level Transformer with Masked Document Representations as Pre-Training for Stance Detection

Matthew Matero, Nikita Soni,
Niranjan Balasubramanian, and **H. Andrew Schwartz**
Department of Computer Science, Stony Brook University
{mmatero, nisoni, niranjan, has}@cs.stonybrook.edu

Abstract

Much of natural language processing is focused on leveraging large capacity language models, typically trained over single messages with a task of predicting one or more tokens. However, modeling human language at higher-levels of context (i.e., sequences of messages) is under-explored. In stance detection and other social media tasks where the goal is to predict an attribute of a message, we have contextual data that is loosely semantically connected by authorship. Here, we introduce Message-Level Transformer (MeLT) – a hierarchical message-encoder pre-trained over Twitter and applied to the task of stance prediction. We focus on stance prediction as a task benefiting from knowing the context of the message (i.e., the sequence of previous messages). The model is trained using a variant of masked-language modeling; where instead of predicting tokens, it seeks to generate an entire masked (aggregated) message vector via reconstruction loss. We find that applying this pre-trained masked message-level transformer to the downstream task of stance detection achieves F1 performance of 67%.

1 Introduction

Generated by people, natural language data inherently spans multiple levels of analysis, from individual tokens, to documents (or messages), and to sequences of messages. While the multi-level aspect is rarely looked at beyond words-to-documents, some work has suggested benefits to modeling language as a hierarchy, such as building document representations from a collection of its sentences or a user vector given a history of their language (Song et al., 2020; Acheampong et al., 2021; Grail et al., 2021; Matero et al., 2019; Ganesan et al., 2021).

We consider stance detection, a message-level task, where the social or personal context in which the message appears (e.g., such as a person's profile) has been shown relevant to capturing the stance of the message (Lynn et al., 2019; Aldayel and Magdy, 2019). However, such work explicitly integrated user- or social-context into the stance model, as a separate component. We ask if there is a more direct integration of user context when processing a target message. To this end, we process the target message as a part of the sequence of messages from the user. This way of using historical language from a person enables us to both model within message information (word-level) and to process the message within the author context (message-level).

While there have been some models that take advantage of hierarchy through words and sequences of messages (Lynn et al., 2020; Yu et al., 2020; Zhao and Yang, 2020) there has been little work in providing generic pre-training routines for large capacity transfer learning style models beyond the word-level. Instead, many of these hierarchical models are either applied directly to a downstream task or, if pre-trained, on an adjacent version of the downstream task. Being able to pre-train general message-level models could enable inclusion of message-level contextual information that is not easily obtainable with task-specific training that is limited in data sizes as compared to larger unlabeled corpora available for modeling at the message-level.

In this study, we propose a hierarchical message-level transformer (MeLT) trained over a novel pre-training routine of *Masked Document Modeling*[1], where the goal is to encode documents in latent space using surrounding contextual documents. We then fine-tune MeLT to a stance detection dataset derived from Twitter as defined in the SemEval 2016 shared task (Mohammad et al., 2016). Our contributions include: (1) introduction of a new pre-training routine for hierarchical message-level transformers[2], (2) demonstration of efficacy of our

[1] In this work a document is a single tweet (referred to as a message)

[2] Code: https://github.com/MatthewMatero/MeLT

pre-training routine for stance detection, and (3) exploratory analysis comparing model size with respect to the number of additional message-level layers and amount of user history leveraged in fine-tuning.

2 Related Work

Our approach is inspired by the success word-to-document level transfer learning has had since popularized by the BERT language model (Devlin et al., 2018). Offering the idea of a "contextual embedding" allows models to properly disambiguate words based on their surrounding context. While other types of language models are also used, usually autoregressive based such as GPT and XLNet (Brown et al., 2020; Yang et al., 2019), many models are variants of the BERT autoencoder style (Liu et al., 2019; Lan et al., 2019).

Both Zhang et al. (2019) and Liu and Lapata (2019) use hierarchical encoder models for summarization tasks. While both models encode sentences using some surrounding context, their pre-training tasks are still that of text generation rather than latent modeling. Yu et al. (2020) encode global context in conversation threads on social media by generating a history vector (concatenated representations of each sub-thread) during the fine-tuning step and Zhao and Yang (2020) propose a capsule network to aggregate fine-tuned word representations to perform automatic stance detection.

Stance detection is an ideal task to develop MeLT because while it is labeled at the message-level, the stance itself is presumed to be held by the author with a history of messages. Previous successful approaches to stance detection have used topic modeling, multi-task modeling via sentiment, multi-dataset training (Lin et al., 2017; Li and Caragea, 2019; Schiller et al., 2021), or user-level information (Lynn et al., 2019; Aldayel and Magdy, 2019). Our work builds on this by using a pre-trained transformer trained to model message representations in latent space across author histories to encode global user knowledge into individual messages.

3 Hierarchical Message Modeling

Messages are made up of individual words that come together to give each other context and meaning. Comparably, a collection of messages can come together to show topics of conversation. Directly encoding the interactions of messages and their underlying words can prove beneficial when modeling language at the document or person-level. For example, processing post history of a social media user within context of their own language.

3.1 Masked-Document Reconstruction

We adapt the masked-language modeling (MLM) approach popularized by use in the BERT model to work for masked documents, rather than words. Namely, we introduce the *masked-document modeling* task, as shown in equation 1, where a message sequence is ordered by created time within a user's history, some messages are selected for masking, and every message is represented as the average of their word tokens.

$$\hat{M}_t = f(M_{t-k}, ..., masked_t, ..., M_{t+k}) + \epsilon \quad (1)$$

Here, $\hat{M}_t$ is the reconstruction of the masked out message M at step t through function f using the contextual messages $M_{t-k}, ..., M_{t-1}, M_{t+1}, ..., M_{t+k}$ with error represented as ϵ. Loss is calculated, as mean-squared-error, against the ground-truth label of the average representation of all words, W_i, that are present in the individual masked message shown in equations 2 and 3. Thus, making the task latent space reconstruction where our model learns to encode messages by rebuilding their local representation using global context.

$$Label = avg(W_0, W_1, ..., W_n) \quad (2)$$

$$Loss = MSE(\hat{M}_t, label) \quad (3)$$

Our masking strategy follows the same rules as introduced in BERT. Specifically, a message has a 15% chance of being selected for masking. Once selected they are then replaced with a message MASK token (80% chance), left unchanged (10% chance), or replaced with a random message vector (10% chance).

3.2 Message-level Transformer (MeLT)

Architecture Description We first select a pre-trained word-level language model on which we build MeLT. This allows us to leverage models that have already shown success in many NLP tasks rather than training from scratch.

After processing messages at the word-level, we average all individual word tokens within a message into a single message vector to build a sequence of message vectors and then select messages for masking. This process and architecture

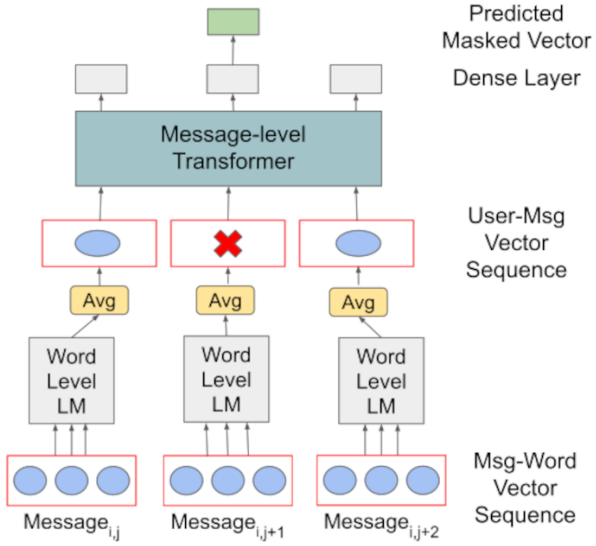

Figure 1: Pre-training architecture of our MeLT model. The bottom layer indicates a collection of a user's individual messages being processed by a word-level language model. Words within individual messages are aggregated as averages and then ordered into a sequence of 768-dimensional message-vectors per user and masking is performed, represented by a red X. Reconstruction loss is then calculated with the predicted masked vector.

is highlighted in figure 1, we refer to models using this setup as a "Message-level Transfomer" (MeLT). Since the loss calculation as described in Eq 3 relies on output from the word-level model itself, that portion of the model is kept frozen during pre-training.

We build 2 versions of MeLT, one with 2 hierarchical layers (2L) and a 6-layer model (6L). After the last transformer layer there is a single dense linear layer which generates the final reconstructed representation of any masked out messages.

These versions of MeLT are built on top of DistilBERT (base) (Sanh et al., 2019) for the following reasons: (1) it is a smaller model (6 layers) allowing more GPU space for message-level layers and (2) while being roughly half the size of the original BERT it still offers upwards of 95% the performance. We also explore an alternate model built-on top of DistilRoBERTa (base) to compare the utility of MeLT applied to other word-level models.

Training Instances For training we set the following restrictions for individual users: (1) we set a max history length of 40 for number of messages per sequence and (2) for users with more than 40 messages they are chunked and processed as separate sequences. Users with fewer than 40 total messages have message-level PAD tokens appended to their sequence. However, users that have multiple sequences will not be assigned a PAD token, if their last sequence falls short of 40 we include the amount of missing messages from their previous sequence.

Model	*F1*	*Prec*	*Recall*	*SemEval F1*
MFC	54	67	78	67
(Zarrella, 2016)	–	–	–	68†
(Zhao, 2020)	–	–	–	78†
DistilBert	60	60	63	63
DistilBert + Hist	63	64	65	68
(Lynn, 2019)	66	–	–	–
MeLT	**67**	**68**	**67**	**73**

Table 1: Evaluation of various methods applied to SemEval stance detection. We report both weighted F1/Prec/Recall and Avg pos/neg F1 as defined in the original shared task. MFC is a most frequent class baseline, DistilBert and DistilBert + Hist represent an average message vector extracted from DistilBERT with or without concatenation of an average vector representing user history, respectively. MeLT is our best performing variant. **Bold** results are found significant with $p < .05$ w.r.t DistilBert + Hist using a paired t-test. (†) indicates a model trained on the original version of the SemEval2016 dataset (4,100 total tweets) which we did not have available due to accounts or messages being deleted on twitter since release.

Dataset For pre-training our model we select users from publicly available tweets that were previously used for other user-level predictions, such as demographic prediction or emotion forecasting (Volkova et al., 2013; Matero and Schwartz, 2020). A subset of data is selected as our pre-training dataset, approximately 10 million tweets sampled from 6 thousand users, resulting in a dataset 1.3 GB in size. We use a limited dataset to highlight the utility of the pre-training routine itself and not rely on "bigger is better" mindset.

4 Stance Detection with MeLT

We use the stance dataset available from the SemEval 2016 shared task (Mohammad et al., 2016). This data includes tweets that were annotated either against, neutral, or favoring of a specific target mentioned within the tweet, across 5 distinct targets in the dataset. However, this data only includes labeled tweets from users and not any history, so we use the extended dataset from Lynn et al. (2019).

During fine-tuning we keep a max history length of 40 and a temporal ordering within sequence. We

Model	*Abortion*	*Atheism*	*Climate*	*Clinton*	*Feminism*	*All(Avg)*
Word-level Pre-train						
DistilBert	60	66	70	58	46	60
DistilBert + Hist	64	62	70	64	54	63
Msg-level Pre-train						
2L MeLT-rand	56	62	61	47	46	54
6L MeLT-rand	56	62	61	47	46	54
2L MeLT + frz word	58	64	66	54	51	59
2L MeLT + unfrz word	**66**	**67**	**74**	58	59	65
6L MeLT + frz word	62	66	68	60	53	62
6L MeLT + unfrz word	**66**	66	71	**67**	**63**	**67**

Table 2: Performance analysis on weighted F1 among all our models across each target within the SemEval dataset. MeLT-rand is our architecture applied directly to the task(no pre-train routine) and frz/unfrz word indicates whether the underlying word-level model was also updated while fine-tuning. **Bold** indicates best in column.

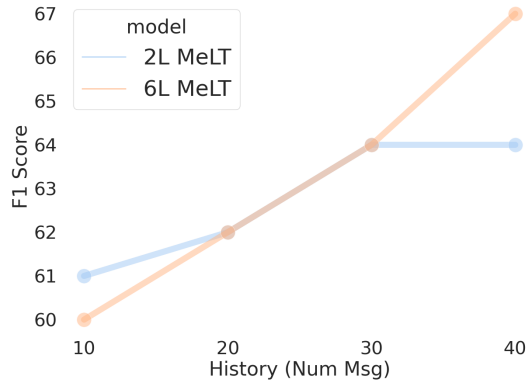

Figure 2: Average weighted-F1 performance across our models when we fine-tune using different amounts of user history. Both size MeLTs improve when more history is available, with a plateau occurring on the 2-layer model.

Model	*F1*	*Prec*	*Rec*
Word-level Pre-train			
DistilRoBERTa	59	55	57
DistilRoBERTa + History	61	68	66
Msg-level Pre-train			
2L MeLT DistilRoBERTa	62	69	66
6L MeLT DistilRoBERTa	**64**	**69**	**69**

Table 3: Evaluation of using a different word-level model for our experiments (DistilRoBERTa). All MeLT variants are fine-tuned with the word-level model unfrozen. While we do not see this version outperform the DistilBERT variant, there are still clear benefits from using MeLT over just the word-level distil-RoBERTa. **Bold** results are found to be significant with $p < .05$ w.r.t DistilRoBERTa + History.

apply a 2-layer feed-forward neural net with a Sigmoid activation on top of our MeLT and leave all message transformer layers unfrozen. Experiments with both frozen and unfrozen word-level layers are also explored. The message vector representation from the top transformer layer of MeLT is used as input into the fine-tuning layers.

5 Results

We show a comparison of our best MeLT model against other approaches in table 1. First, we include a heuristic baseline of most-frequent-class prediction. Next, we compare against fine-tuning our word-level model of choice directly to the downstream task using 2 configurations. The first is using only the message representation, while the second is "+ history" where we concatenate it with the average of 40 recent messages. This allows the model to have a global context within user. We also include the top participant from the shared task Zarrella and Marsh (2016) which uses a different F1 score as defined for the shared task, referred to here as *SemEval F1*[3]. Lastly, we compare our results to the approach of Lynn et al. (2019), from whom we received the extended history dataset, which uses the labeled tweet and a list of accounts the author follows. However, they only report the weighted-F1 score for their best performing model.

We find that fine-tuning DistilBERT directly to the task of stance detection proves difficult, only scoring a modest F1. However when we include some context language from the user, an average representation of their recent language concate-

[3]This F1 score instead reports an average of the F-score for the positive and negative classes. Not directly accounting for neutral predictions.

nated into the fine-tuning layer, there is a noticeable boost in performance highlighting that stance prediction is aided by knowing the context of the message. We find that MeLT can utilize this contextual information best and out-performs other approaches.

Next, we break down the performance of various configurations of our models in table 2 across each target. Here, we compare against a small variant of MeLT (2Layers), randomly initialized MeLTs (No pre-train)[4], and also experiments with frozen and unfrozen word-level models. Ultimately, we find that fine-tuning both the word and message levels simultaneously consistently proves beneficial, likely due to the word model being able to adapt to discourse on Twitter.

We also find that the 2-layer MeLT performs competitively - in figure 2 we show that it performs better or on-par with the large model until 40 messages of history is reached, due to the 2-layer model saturating at history of 30. Suggesting that the larger the model, the more history it can efficiently track.

Lastly, we investigate using a different word-level model for our experiments. We choose DistilRoBERTa, for similar reasons to our original choice of DistilBERT, and apply the same techniques as done with DistilBERT shown in table 3. We find that overall each DistilRoBERTa model achieves lower F1 score than the respective DistilBERT variant. However we find that MeLT still improves over the base word-level model, suggesting that MeLT often will improve the word-level model itself but the word-level model of choice plays an important role in downstream performance. Due to this, it is likely to be beneficial to first evaluate a variety of word-level models on your downstream task and then build on top of the best one with MeLT.

6 Conclusion

With a large number of tasks in NLP that rely on social media as a domain, methods which can model language as a multi-level phenomena, from words to documents to people, can offer a higher-level contextual representation of language. In this work, we presented a new hierarchical pre-training routine that, when fine-tuned to stance detection, outperforms other models utilizing both message and user-level information as well as improves results upon solely using the word-level model on which we build MeLT. We also find that during fine-tuning, it was always beneficial to unfreeze the word layers even though they had to be frozen during pre-training. MeLT can be attached to the top of a word-level language model in order to directly encode sequences of message vectors, thus allowing the modeling of historical context and leading towards a way of approaching language modeling that integrates its personal context.

[4]Both MeLT-rands learn the MFC baseline

7 Acknowledgements

This work was supported in part by a grant from the National Institutes of Health, R01 AA028032-01 and in part by a grant from the National Science Foundation, IIS-1815358.

References

Francisca Adoma Acheampong, Henry Nunoo-Mensah, and Wenyu Chen. 2021. Transformer models for text-based emotion detection: a review of bert-based approaches. *Artificial Intelligence Review*, pages 1–41.

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Abeer Aldayel and Walid Magdy. 2019. Your stance is exposed! analysing possible factors for stance detection on social media. *Proceedings of the ACM on Human-Computer Interaction*, 3(CSCW):1–20.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

et al. Falcon, WA. 2019. Pytorch lightning. *GitHub. Note: https://github.com/PyTorchLightning/pytorch-lightning*, 3.

Adithya V Ganesan, Matthew Matero, Aravind Reddy Ravula, Huy Vu, and H Andrew Schwartz. 2021. Empirical evaluation of pre-trained transformers for human-level nlp: The role of sample size and dimensionality. *arXiv preprint arXiv:2105.03484*.

Quentin Grail, Julien Perez, and Eric Gaussier. 2021. Globalizing BERT-based transformer architectures

for long document summarization. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1792–1810, Online. Association for Computational Linguistics.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.

Yingjie Li and Cornelia Caragea. 2019. Multi-task stance detection with sentiment and stance lexicons. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6300–6306.

Junjie Lin, Wenji Mao, and Yuhao Zhang. 2017. An enhanced topic modeling approach to multiple stance identification. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 2167–2170.

Yang Liu and Mirella Lapata. 2019. Hierarchical transformers for multi-document summarization. *arXiv preprint arXiv:1905.13164*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Veronica Lynn, Niranjan Balasubramanian, and H Andrew Schwartz. 2020. Hierarchical modeling for user personality prediction: The role of message-level attention. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5306–5316.

Veronica Lynn, Salvatore Giorgi, Niranjan Balasubramanian, and H. Andrew Schwartz. 2019. Tweet classification without the tweet: An empirical examination of user versus document attributes. In *Proceedings of the Third Workshop on Natural Language Processing and Computational Social Science*, pages 18–28, Minneapolis, Minnesota. Association for Computational Linguistics.

Matthew Matero, Akash Idnani, Youngseo Son, Salvatore Giorgi, Huy Vu, Mohammad Zamani, Parth Limbachiya, Sharath Chandra Guntuku, and H Andrew Schwartz. 2019. Suicide risk assessment with multi-level dual-context language and bert. In *Proceedings of the Sixth Workshop on Computational Linguistics and Clinical Psychology*, pages 39–44.

Matthew Matero and H. Andrew Schwartz. 2020. Autoregressive affective language forecasting: A self-supervised task. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2913–2923, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Saif Mohammad, Svetlana Kiritchenko, Parinaz Sobhani, Xiaodan Zhu, and Colin Cherry. 2016. Semeval-2016 task 6: Detecting stance in tweets. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 31–41.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Benjamin Schiller, Johannes Daxenberger, and Iryna Gurevych. 2021. Stance detection benchmark: How robust is your stance detection? *KI-Künstliche Intelligenz*, pages 1–13.

Xingyi Song, Johnny Downs, Sumithra Velupillai, Rachel Holden, Maxim Kikoler, Kalina Bontcheva, Rina Dutta, and Angus Roberts. 2020. Using deep neural networks with intra- and inter-sentence context to classify suicidal behaviour. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 1303–1310, Marseille, France. European Language Resources Association.

Svitlana Volkova, Theresa Wilson, and David Yarowsky. 2013. Exploring demographic language variations to improve multilingual sentiment analysis in social media. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1815–1827.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *arXiv preprint arXiv:1906.08237*.

Jianfei Yu, Jing Jiang, Ling Min Serena Khoo, Hai Leong Chieu, and Rui Xia. 2020. Coupled hierarchical transformer for stance-aware rumor verification in social media conversations. Association for Computational Linguistics.

Guido Zarrella and Amy Marsh. 2016. Mitre at semeval-2016 task 6: Transfer learning for stance detection. *arXiv preprint arXiv:1606.03784*.

Xingxing Zhang, Furu Wei, and Ming Zhou. 2019. Hibert: Document level pre-training of hierarchical bidirectional transformers for document summarization. *arXiv preprint arXiv:1905.06566*.

Guangzhen Zhao and Peng Yang. 2020. Pretrained embeddings for stance detection with hierarchical capsule network on social media. *ACM Transactions on Information Systems (TOIS)*, 39(1):1–32.

A Appendix

A.1 Implementation and Hardware Details

Pre-training of all models was performed across 3 TitanXP GPUs(12GB mem each) while fine-tuning was performed on a single TitanXP. All models were implemented using PyTorch (Paszke et al., 2019) with the PyTorch Lightning Add-on (Falcon, 2019).

During pre-training batch size was set to 100 users and fine-tuning was performed using 10. For pre-training runtime was around 2.5 hours per epoch and fine-tuning was a few minutes per epoch. MeLT 2L adds 11,621,632 trainable parameters on top of DistilBERT and MeLT 6L adds 33,677,568, as counted by summing PyTorch tensor.numel() per parameter with gradients turned on[5]. All experiments (pre-training and fine-tuning) use the AdamW Optimizer (Loshchilov and Hutter, 2017) and use random seed set to 1337. Pre-training has a warm-up period of 2,000 steps.

Pre-training is conducted over 5 epochs with checkpoints saved for the epoch that scored the lowest MSE on a holdout development set. The version of the model at that checkpoint is then used for fine-tuning to the stance dataset.

A.2 Hyperparams

All hyperparameters are selected via tuning using the Optuna library (Akiba et al., 2019).

A.2.1 Pre-training

The final set of hyperparameters used for the 6L MeLT model (pre-training) are as follows:

- Learning Rate: 4e-3
- Weight Decay: 0.1
- Dropout: 0.1
- FF dim: 2048
- Embed dim: 768
- Attn Heads: 8
- Epochs: 5 (checkpoint at epoch 2)
- batch size: 100 (users)
- msg seq len: 40 (per user)
- token seq len: 50 (per message)

[5]https://discuss.pytorch.org/t/how-do-i-check-the-number-of-parameters-of-a-model/4325

If any parameter is not mentioned (e.g., Adam Betas) then it uses PyTorch defaults. For pre-training 10 trials were used for parameter tuning. For pre-training only learning rate and weight decay were explored. Learning rate was searched between 5e-4 to 4e-1 and weight decay was set between 1 and 1e-4.

A.2.2 Fine-Tuning

All hyperparameters were chosen based on minimizing loss over a holdout development set for each target over 50 trials. Hyper-parameters that are tuned include learning rate, weight decay, and dropout. Dropout is applied directly to output from MeLT. Learning rate was searched between 6e-6 and 3e-3, weight decay is between 1 and 1e-4, and dropout is 0.0 to 0.05. Additionally, early stopping was also applied as a means of regularization.

The 2-layer FFNN on top of MeLT during fine-tuning has layer 1 of dimension 768 and layer 2 of dimension 384, with Sigmoid between.

A.3 Data

A.3.1 pre-training

The pre-training dataset is comprised of 6,000 users and 9,868,429 messages. For a development set we select 3,000 users from our train set and set aside an additional 20 of their messages, to measure reconstruction loss within these sequences.

A.3.2 fine-tuning

The breakdown of number of examples (labeled messages) across train/dev/test for each target in the SemEval Stance data is shown in table 4. In total we have 3,021 instances with a split of 1658 train, 418 dev, and 945 test across all targets. The original 2016 shared task had 4,100 instances, however due to accounts or messages being deleted over time, we were unable to replicate the complete original dataset and instead used the smaller version available from Lynn et al. (2019).

Target	Train	Dev	Test
Abortion	380	96	207
Atheism	329	83	178
Climate Change	257	65	145
Hilary Clinton	372	94	232
Feminism	320	80	183
Total	1658	418	945

Table 4: Number of examples per target in SemEval data as broken down by split of the data.