Autoencoder Human Language Modeling

Shailen Smith
Stony Brook University
shailen.smith@stonybrook.edu

Abstract

While Language Models (LMs) have become ubiquitous in daily life, most still treat language in isolation, without consideration of the human context in which the language was generated. To address this problem, recent work has proposed the human language modeling (HuLM) task for autoregressive LMs, which seeks to induce dependencies between documents written by the same author. We extend the HuLM task to autoencoding models, taking a simpler method of (1) concatenating a user's most recent documents into a single longer training sequence and (2) extending the context size of the autoencoder to accommodate longer sequences. To test this approach, we continue pre-training a Distil-RoBERTa autoencoder with our proposed human-aware approach on two sets of social media posts, and we compare perplexity against a Distil-RoBERTa model trained with a traditional approach on the same social media data. We find that our human-aware approach is effective on shorter posts, but less effective on longer posts, compared to a traditional approach. Our results demonstrate the potential of HuLM for autoencoders, laying the groundwork for future innovations in human-aware autoencoding architectures.

1 Introduction

Deep Language Models (LMs), which are trained on vast quantities of data to understand and generate human language, can be applied to a wide variety of tasks, including sentiment analysis (Nakov et al., 2013), stance detection (Mohammad et al., 2016), and the creation of generative personal assistants like Chat-GPT (OpenAI, 2024). However, even though language is generated by humans, most LMs neglect to model dependence between documents written by the same author (e.g. a series of blog posts) (Soni et al., 2023). The ability to discern the unique language patterns of individual authors could give language models a more accurate representation of the world, and utilizing these patterns holds significant potential for insights into the authors' changing psychological states (Fleeson, 2001; Mehl and Pennebaker, 2003; Heller et al., 2007).

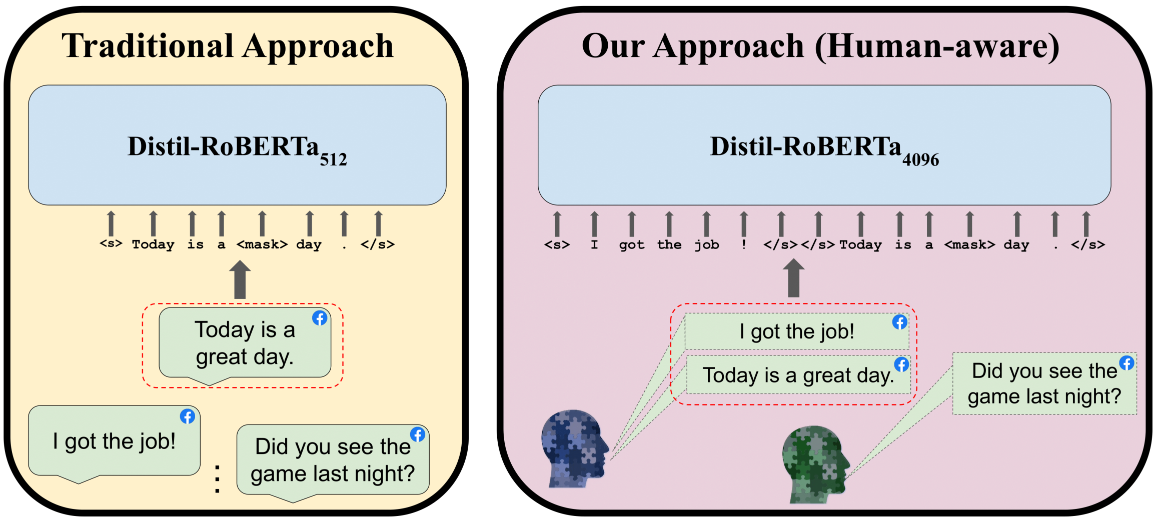
To address this problem, Soni et al. (2022) have proposed the human language modeling (HuLM) task for auto-regressive LMs, which seeks to induce dependence between documents written by the same author. Soni et al. also introduce HaRT (Human-aware Recurrent Transformer), a GPT-2-based decoder that models a recurrent user state **U**, which is updated by the model at regular intervals of training on the author's historical language.

We apply the HuLM task to bidirectional autoencoders, which have strong transfer learning potential for many different non-generative tasks (Devlin et al., 2019). Specifically, we explore the implementation of a basic pre-processing approach of concatenating a certain user's most recent utterances into a single sequence to feed to the autoencoder. As this approach creates significantly longer training sequences, we also extend the maximum context size of our model.

To test this approach, we continue pre-training a human-aware version of the Distil-RoBERTa autoencoder, modified for a longer context size, to train on a user's language from social media posts on Blogger.com and Facebook.com. We assess its performance in terms of perplexity[1] compared to a Distil-RoBERTa model adapted to the social media domain using the same dataset.

Contributions. Our contributions are three-fold: (1) We extend the autoregressive HuLM formulation (Soni et al., 2022) to autoencoders; (2) We provide an empirical analysis of (a) how to best

[1] *Perplexity* is roughly how "confused" the model is by the language in an evaluation dataset. The lower the perplexity, the more confident an autoencoder is in its recreation of the evaluation dataset. See Section 7 for more specific information on the perplexity calculation.



Traditional Approach
Distil-RoBERTa512
<s> Today is a <mask> day . </s>
Today is a great day.
I got the job!
Did you see the game last night?
Our Approach (Human-aware)
Distil-RoBERTa4096
<s> I got the job ! </s></s> Today is a <mask> day . </s>
I got the job!
Today is a great day.
Did you see the game last night?

Figure 1: A comparison example with Facebook.com posts between the traditional approach to masked language modeling for autoencoders and our proposed human-aware approach. The traditional approach only considers the Facebook posts in isolation, while our human-aware approach will see multiple posts from one user combined into a single training sequence. In this example, the additional user history will likely help the model understand that the masked word should be positive after following the positive news of receiving a job. Note that Distil-RoBERTa$_{512}$ refers to the traditional Distil-RoBERTa model with a maximum context size of 512 tokens, whereas Distil-RoBERTa$_{4096}$ refers to our Distil-RoBERTa model with a maximum context size of 4096 tokens, expanded as described in Section 4.5. This example uses the DSEP special token approach, as described in Section 4.4.

use special tokens to separate a single user's concatenated documents and (b) how to best expand the maximum context length of Distil-RoBERTa from 512 tokens to 4096 tokens; (3) We evaluate our human-aware training approach to continued pre-training of two domains of social media posts.

2 Background

2.1 Neural Language Models

Neural language models create continuous, vectorized representations of language. Representing words as vectors allows words with similar meaning to have similar representations. This approach was shown effective in creating meaningful vector representations of words (Bengio et al., 2000); (Bengio et al., 2003), and various neural language models soon showed strong potential on a variety of NLP tasks (Collobert and Weston, 2007); (Collobert and Weston, 2008).

Neural language models continued to develop in the coming years, including the introduction of recurrent neural networks to language modeling (Mikolov et al., 2011) and the creation of the influential word embedding models Word2vec (Mikolov et al., 2013) and GloVe (Pennington et al., 2014), which became the state of the art in many NLP tasks (Jurafsky and Martin, 2000). These models continued to grow larger and more effective until the Transformer (Vaswani et al.) was introduced in 2017, an attention-based model that was shown to scale up very well to very large sets of data, ushering in the era of Large Language Models (LLMs). Instead of single-word vector embeddings, Transformer models output *contextualized representations* of input text that encode information about words within their surrounding context. Transformer models like ELMo (Peters et al., 2018) and BERT (Devlin et al., 2019) defined a new state-of-the-art, establishing the Transformer as the dominant language model architecture in the years to come.

2.2 The Transformer

The Transformer LM architecture primarily consists of a collection of feedforward neural networks and attention layers. Attention allows for further communication of words within a document to understand each others' context (Vaswani et al.). Transformers can have millions or billions of parameters, which can be thought of as knobs and dials that get tweaked with each new document exposed to the model during training.

2.3 Pre-training and Fine-tuning

Transformers are typically first *pre-trained* on a large natural language dataset on either next-word prediction (*causal language modeling* or just *language modeling*) or fill-in-the-blank prediction (*masked language modeling*). Through this resource-intensive process, models learn the rudiments of language and grow their understanding of different words in context. Pre-trained LMs are then often adapted to a specific downstream task using a comparatively small amount of human-labeled data. One example task is stance detection, where the labels are human assessments of whether each message is positive, neutral, or negative for a given topic (Mohammad et al., 2016). This second training process is called *fine-tuning* an LM, and has proven effective in adapting the general language understanding of pre-trained LMs to many different language-related tasks (Naveed et al., 2024).

Transformers are often either be unidirectional or bidirectional. Unidirectional models, called *decoders*, are pre-trained on the language modeling task; bidirectional models, called *autoencoders*, are pre-trained on the masked language modeling task, using past and future context to re-create masked words. Decoders are well-suited for generating new language, whereas autoencoders are more often fine-tuned for a variety of language understanding tasks, including sentiment analysis and stance detection. Popular decoder-only models include GPT-3 (Brown et al., 2020), while popular autoencoders like BERT (Devlin et al., 2019) and its successor RoBERTa (Liu et al., 2019) remain ubiquitous in many areas of large-scale language analysis for their extensive capability for transfer learning for non-generative tasks.

2.4 The Distil-RoBERTa Model

Distil-RoBERTa is an 82M-parameter version of the popular 123M-parameter RoBERTa autoencoder (Liu et al., 2019). Distil-RoBERTa was "distilled" by a process that trains the model to match the output of its "teacher" model, RoBERTa (Sanh et al., 2019). Distil-RoBERTa is on average twice as fast as RoBERTa, and generally outperforms Distil-BERT (Sanh et al., 2019).[2] Distil-RoBERTa has a maximum context size of 512 tokens.

[2]Sanh et al. (2019) mainly describes the training of Distil-BERT, not Distil-RoBERTa. Please visit the Distil-RoBERTa Hugging Face page for more direct information on the model.

2.5 Hyperparameter Tuning

Distil-RoBERTa, as well as other autoencoders, have many hyperparameters that can affect training behavior. *Batch size* determines how many training sequences of language data are processed in parallel during each model forward pass. *Learning rate* controls how much the parameters change after each backward pass. *Weight decay* penalizes larger parameter values as a form of regularization. Before beginning a stage of training, a *hyperparameter tuning* process is typically undergone in which multiple trials of varying hyperparameter values are used to determine the most effective starting hyperparameters, which are then used in the main training (Naveed et al., 2024).

3 Related Work

Our human-aware training approach is related to previous works in domain-adaptive pre-training, user modeling, HuLM, and context length extrapolation.

3.1 Domain-Adaptive Pre-Training

After their initial pre-training phase, LMs can also continue to be pre-trained on a task-agnostic language dataset within a specific domain (e.g. blog posts), effectively "adapting" the LM to better understand language in the specific domain. Gururangan et al. (2020) call this method DAPT (Domain Adaptive Pre-Training), and show that DAPT can improve RoBERTa's performance in various target domains, including biomedical publications and online product reviews. We will use a DAPT-inspired approach in our training of a human-aware model.

3.2 Modeling Social Factors in Language

It has been suggested that while NLP models have performed well on information-related tasks, most are quite limited in their social understanding of human language, and that the NLP community should investigate further methods of including human social factors in language (Hovy and Yang, 2021). This can be seen as part of a broader theme of incorporating extra-linguistic information into language models, helping them take language in its appropriate context. For example, including demographic information (Hovy, 2015) and adapting language models to inferred demographic information (Lynn et al., 2017) have both been effective in various text-classification tasks. Users have also been modeled more directly through their natural language; personalized user embeddings have shown previous success in large-scale social media analysis tasks, being predictive of mental health conditions and similarities between users (Amir et al., 2017). In a similar vein, various methods of personalized language modeling, which seeks to train language models tailored to the style of a particular user's writing, have been shown to both reduce perplexity (King and Cook, 2020) and improve performance on various tasks like texting autocomplete (Jaech and Ostendorf, 2018).

3.3 Human Language Modeling

Soni et al. formalize the notion of Human Language Modeling (HuLM) for decoders as the "task of estimating the probability of a sequence of tokens, $w_{t,1:i}$, while conditioning on a higher order state ($\mathbf{U}_{1:t-1}$) derived from the tokens of other documents written by the same individual." This is summarized in the following objective:

$$Pr(w_{t,i}|w_{t,1:i-1}, \mathbf{U}_{1:t-1}) \tag{1}$$

where $w_{t,i}$ represents the i-th token of the t-th document of the individual, and $\mathbf{U}_j$ represents the j-th user state for the same individual. In addition to its incorporation of social factors and explicit user modeling, HuLM also provides the benefit of avoiding the ecological fallacy of considering two documents from the same user to be independent, as opposed to the other methodologies listed above (Soni et al., 2022).

Soni et al. then create HaRT, a GPT-2 based decoder with an inserted recurrent user-based state. They find significant performance improvements in perplexity and downstream performance on multiple document- and user-level tasks, providing promising evidence that other forms of human-aware LM training, like this paper's approach, could be effective.

3.4 Context Length Extrapolation

Language models typically have a maximum context length allowed for each input. However, extending the maximum context length of a model may allow for the consideration of additional user context over time. Extending the context length of a pre-trained model typically necessitates modifying the LM's positional embeddings that encode the relative position of the tokens to the model. Two types of positional embeddings are *fully learnable* positional embeddings added before the first layer of the model, such as used in RoBERTa (Liu et al., 2019), and *rotary positional embeddings* (RoPE) that interact with the attention layer(s) (Su et al., 2023). Pal et al. (2023) explore a variety of methods to extend the context length of a LLaMa-2 model (Touvron et al., 2023), which uses RoPE. They find a linear interpolation method to be most effective and suggest an exploration into the use of these methods to other forms of positional embeddings. However, the linear interpolation approach for RoPE embeddings does not have a clear analog to Distil-RoBERTa's fully learnable positional embeddings, so we formulate some new approaches to perform context length extrapolation on Distil-RoBERTa (See Section 4.4).

4 Designing an Autoencoding Human Language Model

4.1 Human Language Modeling for Autoencoders

First, let us adapt the HuLM training objective in Equation 1 to autoencoders. A human-aware autoencoder will seek to maximize

$$Pr(w_{t,i}|w_{t,1:i-1;i+1:n}, U)$$

where $w_{t,i}$ is the i-th token of the t-th document of user U. Note that U in this formulation does not represent an *explicit* user representation, as in Equation 1; we make this generalization because our approach only *implicitly* models each user.

4.2 Using Distil-RoBERTa

We choose the Distil-RoBERTa autoencoder (Sanh et al., 2019) for our experiment based on its continued relevance in the field and its smaller size to allow for faster training.

4.2.1 A Note on Notation

As we will be discussing different versions of Distil-RoBERTa trained with various maximum context sizes and datasets, let us have DistBERTa$_n^{\text{DATA}}$ refer to a pre-trained Distil-RoBERTa model with maximum context size n that has been pre-trained further on the train split of the dataset DATA. DistBERTa$_n$, with no superscript, will refer to the pre-trained Distil-RoBERTa model with maximum context size n that has not received additional pre-training. (With this notation, the traditional Distil-RoBERTa model with no modifications or extra pre-training is DistBERTa$_{512}$.)

Name	Description	# Tokens	# Users	# Posts
BLOGS	Blogger.com posts from 2001 to 2004	51.9M	19.0k	252k
FB	Facebook posts from 2010 to 2020	4.10M	1.11k	148k
FBLOGS	The above two datasets combined	56.0M	20.1k	400k

Table 1: Datasets used in our main experiment to adapt Distil-RoBERTa to the Blogger.com and Facebook.com post domains.

4.3 Our Human-Aware Approach

To modify the Distil-RoBERTa training process to be human-aware, we take each of a user's most recent documents and concatenate them into one training sequence, creating one longer sequence per user. This concatenation of all user documents into one extended training sequence (a) gives the model an implicit assumption that each sequence corresponds to one user's text and (b) allows the model to understand the history of a user's experiences over time through their language.

However, this concatenation necessitates an exploration into (1) how to best use special tokens to separate a single user's concatenated documents and (2) how to best expand the maximum context length of Distil-RoBERTa from 512 tokens to 4096 tokens (see Appendix .2) in order to fit more of a user's documents in one sequence.

4.4 Treatment of Special Tokens

Distil-RoBERTa uses the special tokens `<s>` and `</s>` to help mark document boundaries for each sequence. However, in the traditional approach, these special tokens are used differently depending on whether there are single documents or pairs of documents in each training sequence. If there are single documents, a training sequence takes the form `<s><d1></s>`. The `<s>` and `</s>` tokens signify the beginning and end of the document to the model. When processing pairs of documents, an sequence takes the form `<s><d1></s></s><d2></s>`. Here, `<s>` and `</s>` signify the respective beginning and end of the sequence, but `</s></s>` is additionally used as a document separator (Liu et al., 2019).

However, our human-aware approach will often create training sequences with more than two documents, and in this case there is no clear standard for what special token grammar is appropriate. For our human-aware method, we consider two plausible approaches, which we call (1) *document start-stop* or DOCSS and (2) *double separator* or DSEP. The makeup of an sequence for a user with most recent documents $\{d_1, \ldots, d_n\}$ is shown in Table 2.

Name	Special Token Usage
DOCSS	`<s><d1></s><s><d2>. . . </s>`
DSEP	`<s><d1></s></s><d2>. . . </s>`

Table 2: Naming special token approach candidates. In the first approach, `<s>` is used to signify the start of a document d_i, and `</s>` the end, hence our name "document start-stop" (DOCSS). In the second approach, `<s>` is used to signify the beginning of the entire sequence, and `</s></s>` is used as a document separator, hence our name "double separator" (DSEP).

The DOCSS approach is based off the form of single-document training described above. It has the additional benefit of being interpretable as XML, as every `<s>` tag is eventually "closed" with a `</s>` tag. The DSEP approach extrapolates from the document-pair special token grammar for RoBERTa, treating `</s></s>` as a document separator between all d_i in a certain sequence. This approach has the advantage of using `</s></s>` as a document separator, which RoBERTa has already seen.

4.5 Extending Context Length

In order to accommodate more user documents within a training sequence for our human-aware approach, we increase Distil-RoBERTa's maximum context size from 512 to 4096 tokens.[3] This involves the manipulation of Distil-RoBERTa's positional embeddings initialization, which we will call *pos-init* as a shorthand.[4]

Distil-RoBERTa's loaded positional embeddings are a fully learnable (512 x 768) matrix P, where the i-th row is a 768-dimensional vector that corresponds to the i-th token position. The number of rows in P is equal to the maximum context size of the model. So, to create DistBERTa$_{4096}$, we then must replace P with a (4096 x 768) positional

[3]See Appendix .2 for more information on the choice of 4096 as our extended maximum context size.

[4]*pos-init* as a shorthand for *positional embeddings initialization* is our own construction as an aid for its repeated reference in this paper. It is not an established shorthand.

Stat	BLOGS	FB	FBLOGS
mean	265	27.1	116
std	599	4.41	38.5
min	3	3	3
1%	5	3	3
25%	46	11	14
50%	141	18	27
75%	329	31	89
99%	1730	158	1170
max	178k	6890	178k

Table 3: Document lengths of full train split (# tokens) for each domain. Posts in FB are typically much shorter than the posts in BLOGS.

embeddings matrix, which we will call P'. We consider three candidate approaches for initializing P':

1. NO-LOAD: Randomly initialize all 4096 rows of P'.
2. LOAD-512: Set the first 512 rows of P' to be P and initialize the remaining 3584 rows randomly.
3. LOAD-REPEAT: Set P' to be 8 copies of P repeatedly stacked vertically.

It is worth noting that however P' is initialized, its weights will be updated during the extended pre-training to better reflect the respective positions of the tokens.

5 Data

We train on a collection of Blogger.com and Facebook.com social media posts from users over the past 25 years. The Blogger.com posts are from the Blog Authorship Corpus (Schler et al., 2006), collected in August 2004 to analyze blogging behavior across different demographics; we will call this dataset BLOGS. The Facebook.com posts are from U.S. restaurant and hospitality workers as part of a longitudinal analysis of affect and drinking habits based on social media data (Matero et al., 2024); we will call this dataset FB. We concatenate the two datasets, and we call the combined dataset FBLOGS. Table 1 provides an overview of the data. As seen in Table 3, the Facebook posts are generally much shorter than the Blogger.com posts.

5.1 Data Pre-Processing

Each blog or FB post in FBLOGS has a corresponding date of post and user ID. We make a 90-5-5 train/dev/test split, such that the posts of 90% of users are placed in the train set, while 5% of users are placed in the dev and test sets. We then sort by date within users, so that each user has their rows of posts grouped together, with most recent posts first.

We then tokenize the text in the posts with the RoBERTa tokenizer (Liu et al., 2019). We remove the top 1% of users by median post length[5], making the assumption that posters at the highest level of verbosity write in a qualitatively different manner from others. We also remove users who have fewer than 500 tokens across all of their posts. Let us refer to the dataset after these processing steps as FBLOGS-TR.

We then create a new dataset by concatenating each user's most recent posts together, creating one row (training sequence) per user. Let us call this dataset FBLOGS-HA. We truncate each user's row at 4096 tokens to match the maximum context size of DistBERTa$_{4096}$; if this truncation is necessary, we also replace the 4096th token in the row with `</s>` to mark the new end of the training sequence. We discard the extra tokens from each user from FBLOGS-HA, and we discard the same tokens from FBLOGS-TR.

Algorithm 1 BREAKUP_512

```
1:  procedure BREAKUP_512($R$)
2:      $m \leftarrow$ length of $R$
3:      $R' \leftarrow$ empty set
4:      if $m \geq 512$ then
5:          $k \leftarrow \lfloor \frac{m}{512} \rfloor$
6:          for $i \leftarrow 0$ to $k$ do
7:              add $R[512i:512(i+1)]$ to $R'$
8:          add $R[512k:m]$ to $R'$
9:      else
10:         $R' \leftarrow \{R\}$
11:     return $R'$
```

We want FBLOGS-TR to have a maximum length of 512 tokens per training sequence without truncation or removal of posts. To do this, consider a row R of tokens $R[i]$ in FBLOGS-TR[6]. We apply Algo-

[5]We calculate post length by # tokens, not # words. Tokens can include words, subwords, and punctuation.

[6]Note that in FBLOGS-TR, a row R corresponds to a single blog or FB post.

rithm 1, BREAKUP_512, to each row R, replacing R with the set of rows R', with each element of R' containing at most 512 tokens.

This completes the pre-processing of our datasets FBLOGS-TR and FBLOGS-HA.

5.2 A Small User Sample

For use in the special token experiment (see Section 6.1), we take a 5% user sample S from the FBLOGS-HA train and dev splits and create two copies, the first with the DOCSS special token approach, and the second with the DSEP special token approach (see Section 4.4). We will call these two sample datasets S-DOCSS and S-DSEP, respectively.

6 Methodology

We seek to perform domain-adaptive pre-training on two Distil-RoBERTa models on the masked language modeling objective, one using a human-aware training approach, and one using the traditional approach. Using our notation from Section 4.2.1, the human-aware approach will produce $\text{DistBERTa}_{4096}^{\text{FBLOGS-HA}}$, while the traditional approach will produce $\text{DistBERTa}_{512}^{\text{FBLOGS-TR}}$.

6.1 Human-Aware Modeling Decisions

Before we can train DistBERTa_{4096} on FBLOGS-HA, we must choose a special token approach and pos-init approach for our DistBERTa_{4096} model. To do so, we run small comparison experiments on the randomly chosen 5% user sample S from FBLOGS-HA dataset (see Section 5.2). We set DOCSS and LOAD-REPEAT as our default respective approaches, and potentially change them based on the results of the below modeling experiments.

First, to choose a special token approach, we train $\text{DistBERTa}_{4096}^{S\text{-DOCSS}}$ and $\text{DistBERTa}_{4096}^{S\text{-DSEP}}$. We separately tune learning rate and weight decay on both models for 10 epochs per trial, and we observe dev set perplexity for the best hyperparameter configurations of each. We use a learning rate range of [3e-5, 3e-4] and a weight decay range of [0.001, 0.1]; we run 20 trials total.[7] We obtain a learning rate of 1.2e-4 and weight decay of 7.5e-3 for the DSEP approach, and a learning rate of 1.2e-4 and weight decay of 7.4e-3 for the DOCSS approach. We use the superior special token approach in all future human-aware training runs; Let S' be S with the superior special token approach.

Second, to choose a pos-init approach, we train three $\text{DistBERTa}_{4096}^{S'}$ models for each of the NO-LOAD, LOAD-512, and LOAD-REPEAT approaches. We separately tune learning rate and weight decay on all three models using the same approach as for the special token hyperparameter tuning; we obtain respective learning rates of 3e-5, 3e-5, and 1.2e-4, and weight decays of 8e-2, 8e-2, and 8e-2 for the three approaches NO-LOAD, LOAD-512, and LOAD-REPEAT, in that order. We observe dev set perplexity on the three models with the best hyperparameter configurations. We use the superior initialization approach to create DistBERTa_{4096} in all future human-aware training runs.

6.2 The Main Experiment

6.2.1 Training the Human-Aware Model

We further pre-train DistBERTa_{4096} on the full FBLOGS-HA train split. We reuse the learning rate and weight decay values tuned in the pos-init experiment for the best approach. We train for 50 epochs with early stopping on the condition that the FBLOGS-HA dev split loss does not improve for 5 epochs in a row. This creates $\text{DistBERTa}_{4096}^{\text{FBLOGS-HA}}$, our trained human-aware model.

6.2.2 Training the Traditional Model

To compare against the human-aware model, we further pre-train DistBERTa_{512} on the full FBLOGS-TR train split. We tune the effective batch size, learning rate, and weight decay hyperparameters, resulting in an effective batch size of 2048, a learning rate of 2e-4, and a weight decay of 7.8e-2. We train for 50 epochs with the same early stopping condition as in Section 6.2.1. This creates $\text{DistBERTa}_{512}^{\text{FBLOGS-TR}}$, our trained traditional model.

7 Evaluation

For the smaller human-aware modeling experiments in Section 6.1, we simply observe the dev set perplexity computed by the Hugging Face Trainer. For the main experiment comparing the traditional model $\text{DistBERTa}_{512}^{\text{FBLOGS-TR}}$ against the human-aware model $\text{DistBERTa}_{4096}^{\text{FBLOGS-HA}}$, for each model, we calculate document-level and user-level perplexity on the BLOGS and FB domains for both dev and test splits. We also compute perplexity in this manner for an out-of-the-box pre-trained Distil-RoBERTa model (in our notation, DistBERTa_{512})

[7]To choose hyperparameters for each trial, we use the default configuration of `hyperparameter_search()` from the Hugging Face `trainer` with an Optuna (Akiba et al., 2019) backend.

Approach	BLOGS		FB	
	dev	test	dev	test
DistBERTa$_{512}$	17.76	19.28	29.95	37.44
DistBERTa$_{512}^{\text{FBLOGS-TR}}$	5.95	6.15	8.95	10.66
DistBERTa$_{4096}^{\text{FBLOGS-HA}}$ **(human-aware)**	6.51	6.49	8.74	9.64

Table 4: Document-level perplexity values for different training approaches and data domains. The traditional model DistBERTa$_{512}^{\text{FBLOGS-TR}}$ slightly outperforms the human-aware model DistBERTa$_{4096}^{\text{FBLOGS-HA}}$ in the BLOGS domain, but the human-aware model has slightly better performance in the FB domain. We note that this experiment slightly favors the traditional approach due to the batch size difference mentioned in 8.2.1. Both the traditional and human-aware training approaches show clear improvement over the out-of-the-box Distil-RoBERTa model.

Approach	BLOGS		FB	
	dev	test	dev	test
DistBERTa$_{512}$	20.07	23.07	32.65	38.09
DistBERTa$_{512}^{\text{FBLOGS-TR}}$	6.41	6.62	9.78	11.04
DistBERTa$_{4096}^{\text{FBLOGS-HA}}$ **(human-aware)**	7.33	7.22	9.34	10.39

Table 5: User-level perplexity values for different training approaches and data domains. The traditional model DistBERTa$_{512}^{\text{FBLOGS-TR}}$ slightly outperforms the human-aware model DistBERTa$_{4096}^{\text{FBLOGS-HA}}$ in the BLOGS domain, but the human-aware model has slightly better performance in the FB domain. We note that this experiment slightly favors the traditional approach due to the batch size difference mentioned in 8.2.1. Both the traditional and human-aware training approaches show clear improvement over the out-of-the-box Distil-RoBERTa model.

in order to assess the effectiveness of the continued pre-training.

7.1 Negative Log Likelihood Calculation

Let $\{U_1 \dots U_m\}$ be the set of users in FBLOGS-TR and FBLOGS-HA, and let $R_{k,t}$ be the t-th document of user U_k. Then let $w_{k,t,i}$ represent the i-th masked token of document $R_{k,t}$, and let $p^M_{k,t,i}$ represent the probability[8] of this token being correctly guessed by model M. Then, we calculate negative log-likelihood (NLL) for document $r_{k,t}$ and model M as follows:

$$NLL(R_{k,t}, M) = -\frac{\sum_i^{n_{k,t}} \log(p^M_{k,t,i})}{n_{k,t}}$$

where $n_{k,t}$ is the number of masked tokens in $R_{k,t}$. Note that NLL is calculated from the probabilities of the model output, as opposed to *cross-entropy loss*, which is calculated from the log-probabilities (logits) of the model output. Also note that we mask the same tokens in FBLOGS-TR and FBLOGS-HA to ensure an accurate comparison.

[8]We extract these probabilities $p^M_{k,t,i}$ through the insertion of custom behavior to the `predict()` function of the Hugging Face Trainer. For more implementation details, please contact the author.

7.2 Perplexity Calculation

We compute perplexity at the document-level and the user-level for each model. We compute *document-level perplexity* as $\exp(\overline{NLL})$, where $\overline{NLL}$ is the mean NLL across all documents in the dataset. We compute *user-level perplexity* as

$$\frac{\sum_k^m \exp(\overline{NLL_k})}{m}$$

where $\overline{NLL_k}$ is the mean NLL across all documents authored by user U_k.

8 Results

8.1 Special Tokens and Pos-init

Table 6 compares the performance of the DOCSS and DSEP special token approaches on the 5% user sample as described in Section 4.4. The DSEP approach achieves a slightly better performance than DOCSS, suggesting that `</s></s>` is a meaningful document separator to Distil-RoBERTa even with more than two documents per sequence.

Table 7 assesses the performance of the NO-LOAD, LOAD-512, and LOAD-REPEAT pos-init approaches on the same 5% user sample. LOAD-REPEAT significantly outperforms the other approaches; NO-LOAD and LOAD-512 both seem to be hampered by the presence of positional embedding weights that have received no pre-training,

Special Token Approach	Dev (*ppl*)
DSEP	**11.72**
DOCSS	12.03

Table 6: Comparing special token approaches on a sample of 5% of users from the train split of FBLOGS-HA. The DSEP approach outperforms the DOCSS approach, potentially suggesting that Distil-RoBERTa values the use of `</s></s>` as a document separator within a training sequence.

Init. approach	Dev (*ppl*)
NO-LOAD	473.0
LOAD-512	215.3
LOAD-REPEAT	**10.51**

Table 7: Comparing pos-init approaches on a sample of 5% of users from the train split of FBLOGS-HA. LOAD-REPEAT is the only approach that learns the sample set well. This implies that the randomly initialized positional embeddings present in the NO-LOAD and LOAD-512 approaches significantly hindered Distil-RoBERTa's understanding, even with 20 epochs to adjust themselves. It is possible, however, that these randomly initialized embeddings would improve significantly if given more than 20 epochs to train.

even after having 20 epochs to adapt. This potentially speaks to the importance of the large quantities of data necessary for pre-training from scratch to create useful fully learnable positional embeddings.

Based on these results, we adopt the DSEP and LOAD-REPEAT approaches for our human-aware model, $\text{DistBERTa}^{\text{FBLOGS-HA}}_{4096}$.

8.2 Main Experiment

Table 4 gives the document-level performance of the traditional model $\text{DistBERTa}^{\text{FBLOGS-TR}}_{512}$ and the human-aware model $\text{DistBERTa}^{\text{FBLOGS-HA}}_{4096}$ on the BLOGS and FB domains, and Table 5 does the same for user-level performance. At both document-level and user-level, the traditional model slightly outperforms the human-aware model on BLOGS, but the human-aware model achieves a slightly better performance on FB. This result suggests that our human-aware training approach fares well in a domain with shorter documents, perhaps because a concatenated training sequence for a user can contain more posts.

8.2.1 A Note On Generalization

We also note that, at both document- and user-level and across domains, our human-aware approach appears to generalize better from the dev to test split, often actually achieving a slightly better performance on the test split than the corresponding dev split. However, the decreased generalizability of the traditional approach could also result from the increased involvement of its dev set compared to the human-aware approach; the traditional model uses the dev set to help choose the most effective batch size, whereas the human-aware approach is fixed with a batch size of 1 due to memory limitations (See Appendix 8). We will train a traditional model using a batch size that gives the traditional and human-aware models an equal number of gradient updates; because of time limitations, we do not include this result here. In any case, a more important test of generalizability will come when fine-tuning these trained models to downstream tasks like sentiment analysis or stance detection.

9 Limitations

We note that this experiment slightly favors the traditional approach due to the batch size difference mentioned in 8.2.1. Since the traditional model training was able to optimize for batch size, but the human-aware model batch size was fixed at 1, the traditional model may have had a batch size most conducive to effective training. Future work should fix the batch sizes for each approach such that the total number of gradient updates during training is comparable.

Additionally, our analysis does not include an assessment of the performance of our trained human-aware model on FBLOGS-TR, the combined dataset without the user concatenation and with a maximum context size of 512. Assessing this performance could provide a more accurate understanding of how the human-aware and traditional models compare in their typical real-world application.

10 Conclusion

Despite the increasing ubiquity of language models in our complex and social world, language models still typically treat language as if it was generated in isolation (Soni et al., 2023). We extend the Human Language Modeling (HuLM) approach proposed by Soni et al. to autoencoder models, and we propose a simple human-aware training approach that concatenates each of a user's most recent posts together into one training sequence with an extended context size. To test this approach, we adapt two Distil-RoBERTa models to the FBLOGS dataset, one using the traditional method, and one using our human-aware method. In an assessment of document- and user-level perplexity, we find that the traditional approach fares better on the longer BLOGS posts, whereas our human-aware approach performs better on the short FB posts.

Our results lay the groundwork for others to answer important questions about the effectiveness of human-aware autoencoders. For example, one could fine-tune these two trained models on a downstream social task like sentiment analysis or stance detection, as it is possible that the trained human-aware model generalizes better to such tasks in ways that are not reflected by a perplexity calculation. In addition, our approach of user concatenation and context expansion is just one simple way in which the training of an autoencoder can be modified to be human-aware; other potential architectural changes include time-based positional embeddings, as well as an implementation of an explicit recurrent user-based state, as is done in Soni et al. (2022).

Our results additionally demonstrate the potential of human-aware training approaches to improve the performance of autoencoders without any significant increase in # parameters, training time, or amount of data. This approach can be applied to the training process of any autoencoder with fully learnable positional embeddings, provided that user and date information is available for the training data. We expect further innovations of training human-aware autoencoders to be fruitful.

Ethics Statement

As Soni et al. note in their discussion of the ethical considerations of HuLM, (1) models that better understand the human context of their language can also be exploited for various unfair practices, such as the non-consensual use of user data for tailored advertisement recommendation, and (2) considerations of privacy must be heightened when releasing human-aware models, as vulnerable models likely have a higher potential of exposing user-specific information than traditional, human-unaware models.

More considerations will be necessary when considering the public release of a human-aware autoencoder.

Acknowledgements

This work was performed in partial fulfillment of the Stony Brook University Honors College Undergraduate Senior Thesis, under the supervision of PhD mentors Adithya V Ganesan and Nikita Soni, and faculty advisor H. Andrew Schwartz. I thank my mentors and advisor, as well as the rest of the HLAB, for their continuous support, mentorship, and community.

This work was supported in part by a grant from the NIH-NIAAA (R01 AA028032) and the Intelligence Advanced Research Projects Activity (IARPA), via the HIATUS Program contract (585968).

References

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. Optuna: A next-generation hyperparameter optimization framework.

Silvio Amir, Glen Coppersmith, Paula Carvalho, Mário J. Silva, and Byron C. Wallace. 2017. Quantifying mental health from social media with neural user embeddings.

Y. Bengio, Réjean Ducharme, and Pascal Vincent. 2000. A neural probabilistic language model. volume 3, pages 932–938.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. 2003. A neural probabilistic language model. *J. Mach. Learn. Res.*, 3:1137–1155.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners.

Ronan Collobert and Jason Weston. 2007. Fast semantic extraction using a novel neural network architecture. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 560–567, Prague, Czech Republic. Association for Computational Linguistics.

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: deep neural networks with multitask learning. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, page 160–167, New York, NY, USA. Association for Computing Machinery.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

William Fleeson. 2001. Toward a structure-and process-integrated view of personality: Traits as density distributions of states. *Journal of personality and social psychology*, 80(6):1011.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks.

Daniel Heller, Jennifer Komar, and Wonkyong Beth Lee. 2007. The dynamics of personality states, goals, and well-being. *Personality and Social Psychology Bulletin*, 33(6):898–910.

Dirk Hovy. 2015. Demographic factors improve classification performance. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 752–762, Beijing, China. Association for Computational Linguistics.

Dirk Hovy and Diyi Yang. 2021. The importance of modeling social factors of language: Theory and practice. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 588–602, Online. Association for Computational Linguistics.

Aaron Jaech and Mari Ostendorf. 2018. Personalized language model for query auto-completion. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 700–705, Melbourne, Australia. Association for Computational Linguistics.

Daniel Jurafsky and James H. Martin. 2000. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*, 1st edition. Prentice Hall PTR, USA.

Milton King and Paul Cook. 2020. Evaluating approaches to personalizing language models. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 2461–2469, Marseille, France. European Language Resources Association.

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.

Veronica Lynn, Youngseo Son, Vivek Kulkarni, Niranjan Balasubramanian, and H. Andrew Schwartz. 2017. Human centered NLP with user-factor adaptation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1146–1155, Copenhagen, Denmark. Association for Computational Linguistics.

Matthew Matero, Huy Vu, August Nilsson, Syeda Mahwish, Young Min Cho, James McKay, Johannes Eichstaedt, Richard Rosenthal, Lyle Ungar, and H. Andrew Schwartz. 2024. Using daily language to understand drinking: Multi-level longitudinal differential language analysis. In *Proceedings of the 9th Workshop on Computational Linguistics and Clinical Psychology (CLPsych 2024)*, pages 133–144, St. Julians, Malta. Association for Computational Linguistics.

Matthias R Mehl and James W Pennebaker. 2003. The sounds of social life: a psychometric analysis of students' daily social environments and natural conversations. *Journal of personality and social psychology*, 84(4):857.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality.

Tomáš Mikolov, Stefan Kombrink, Lukáš Burget, Jan Černocký, and Sanjeev Khudanpur. 2011. Extensions of recurrent neural network language model. In *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5528–5531.

Saif Mohammad, Svetlana Kiritchenko, Parinaz Sobhani, Xiaodan Zhu, and Colin Cherry. 2016. A dataset for detecting stance in tweets. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 3945–3952, Portorož, Slovenia. European Language Resources Association (ELRA).

Preslav Nakov, Sara Rosenthal, Zornitsa Kozareva, Veselin Stoyanov, Alan Ritter, and Theresa Wilson. 2013. SemEval-2013 task 2: Sentiment analysis in Twitter. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 312–320, Atlanta, Georgia, USA. Association for Computational Linguistics.

Humza Naveed, Asad Ullah Khan, Shi Qiu, Muhammad Saqib, Saeed Anwar, Muhammad Usman, Naveed Akhtar, Nick Barnes, and Ajmal Mian. 2024. A comprehensive overview of large language models.

OpenAI. 2024. Gpt-4 technical report.

Arka Pal, Deep Karkhanis, Manley Roberts, Samuel Dooley, Arvind Sundararajan, and Siddartha Naidu. 2023. Giraffe: Adventures in expanding context lengths in llms.

Jeffrey Pennington, Richard Socher, and Christopher 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, Doha, Qatar. Association for Computational Linguistics.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations.

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

Jonathan Schler, Moshe Koppel, Shlomo Engelson Argamon, and James W. Pennebaker. 2006. Effects of age and gender on blogging. In *AAAI Spring Symposium: Computational Approaches to Analyzing Weblogs*.

Nikita Soni, Matthew Matero, Niranjan Balasubramanian, and H. Andrew Schwartz. 2022. Human language modeling. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 622–636, Dublin, Ireland. Association for Computational Linguistics.

Nikita Soni, H Andrew Schwartz, João Sedoc, and Niranjan Balasubramanian. 2023. Large human language models: A need and the challenges. *arXiv preprint arXiv:2312.07751*.

Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. 2023. Roformer: Enhanced transformer with rotary position embedding.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need.

.1 Implementation Details

Please email the author for further information on implementation.

.2 Memory

We train on one A6000 GPU with 48 GiB of RAM; we seek to understand the maximum possible context length that could fit one batch into the GPU's RAM. We set the batch size to 1 to minimize its influence. Table 8 shows memory usage and training time of various larger context sizes. Based on the results, we choose a context size of 4096 tokens, as it fits within our available memory while taking less than 5 days to train a full model.

Batch size	Context size	RAM used (GiB)
1	512	3.2
1	1024	6
1	2048	11.8
1	**4096**	**38**
1	6144	36.2
1	7168	44.0
1	7680	48.3
64	**512**	**45.0**

Table 8: Est. memory used during training on 1 Cronus GPU.