Masked Measurement Prediction: Learning to Jointly Predict Quantities and Units from Textual Context

Anonymous ACL submission

Abstract

Physical measurements constitute a large portion of numbers in academic papers, engineering reports, and web tables. Current benchmarks fall short of properly evaluating numeracy of pretrained language models on measurements, hindering research on developing new methods and applying them to numerical tasks. To that end, we introduce a novel task, Masked Measurement Prediction (*MMP*), where a model learns to reconstruct a number together with its associated unit given masked text. *MMP* is useful for both training new numerically informed models as well as evaluating numeracy of existing systems. To address this task, we introduce a new **Ge**nerative **M**asked **M**easurement (*GeMM*) model that jointly learns to predict numbers along with their units. We perform fine-grained analyses comparing our model with various ablations and baselines. We use linear probing of traditional pretrained transformer models (RoBERTa) to show that they significantly underperform jointly trained number-unit models, highlighting the difficulty of this new task and the benefits of our proposed pre-training approach. We hope this framework accelerates the progress towards building more robust numerical reasoning systems in the future.[1]

1 Introduction

Many natural language processing tasks require a deep understanding of numbers – for example, reading comprehension (Ran et al., 2019), textual entailment (Sammons et al., 2010; Roy, 2017) and hybrid table tasks such as fact-verification (Chen et al., 2020) or question answering (Chen et al., 2021). Masked number prediction (*MNP*) is a popular pretraining objective to imbue language models with numerical understanding and evaluate existing models for their numerical capacity.

[1]We will release our trained models and data-splits upon acceptance on Github.

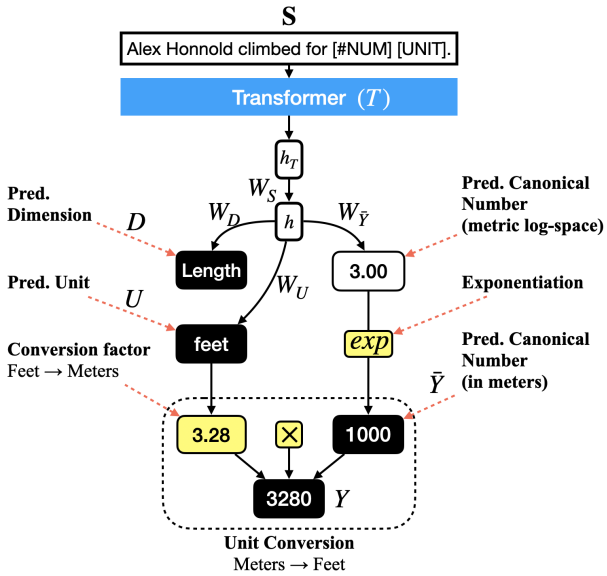


Figure 1: We present the Masked Measurement Prediction (*MMP*) task where the model predicts the dimension, unit and real-valued number. We also show the model architecture of **Ge**nerative **M**asked **M**easurement model (*GeMM*), the model we propose to perform *MMP*. We display the fixed operations used during unit conversion in yellow. In black, we show the different components of the model's prediction.

As an example of *MNP*, given the sentence *"Cats have [#NUM] paws."* a model learns to predict the number *4*. While appropriate for numerical commonsense, *MNP* is deficient when it is used to predict measurements. *Measurements*, such as *2 meters* or *13.2 square miles*, are a special class of particularly common numbers in text that have a well-defined and typed system of *units*. Given a simple question: *"How long did Alex Honnold climb for?"*, a single number alone is an insufficient answer since it is meaningless without the unit. Answers like *1000 meters* or *4 hours* could both suffice.

Current *MNP* systems do not jointly reason about numbers *with* units. It is reasonable to expect that pretrained models like BERT could lever-

age information of units directly as text without any special treatment. However, in preliminary experiments we find that this yields poor numerical abilities (see Appendix B). Furthermore, including units as text directly raise more questions: should we evaluate using all units (*meters, feet, inches*)? Should we equally weight across the units? Current models have no opinion about which unit is appropriate because they are not required to make unit predictions during training. Together, this indicates that current training objectives do not capture sufficient representations of measurements and that a direct application of *MNP* to evaluate numeracy of measurements is ill-suited.

To address these shortcomings, we propose the more challenging task of Masked Measurement Prediction (*MMP*) along with a new model. In this task, a model must reconstruct both the number together with the correct unit. In Figure 1 we show how in a *MMP* setting our model generates a dimension ("Length"), a number in metric log-space ("3.00"), the unit ("feet") and then uses the conversion factor ("3.28") to deterministically output the full measurement ("3280 feet"). This example illustrates a key distinction in that our model is flexible and can generate *non-metric measurements* (feet) but evaluates numerical prediction in canonical units (meters).[2]

MMP is useful for two reasons: 1) as a way to *train* models to give them better numeracy 2) as a new kind of *evaluation* that allows for a much more fine-grained analysis of reasoning over numerical quantities. The task of measurement estimation decouples the different aspects of numeracy allowing for a more interpretable and thorough analysis of numerical reasoning. We introduce a new evaluation benchmark for *MMP* based on WikiConvert (*WiCo*) (Thawani et al., 2021a), a large scale dataset of English Wikipedia sentences with ground truth measurement annotations. We compare the performance of our models on their ability to accurately predict the dimension, unit, and value of a measurement. We employ a large pretrained transformer model as our textual encoder and examine the performance of different discriminative, generative, and latent variable models along with several ablations. Our contributions are as follows:

- We introduce a novel challenging task *MMP* for pretraining and evaluating numeracy.
- We show that linear probing of existing pretrained models on MMP *significantly underperforms* fully finetuned models.
- We train a model that reasons jointly about numbers and units which predicts numbers 8.1 times more accurately than the probed pretrained models.
- We find our best performing generative model outperforms human annotators on two evaluations, achieving 7.4-7.8% better dimension accuracy and 33.5-39.9% better unit accuracy. Furthermore, this model predicts a number **closer** to ground truth than our annotators 66.2-78.8% of the time.

[2]Our metric of choice described in Equation 2 is invariant to the specific choice of canonical unit i.e., *log-mae* in meters is equal to *log-mae* in feet.

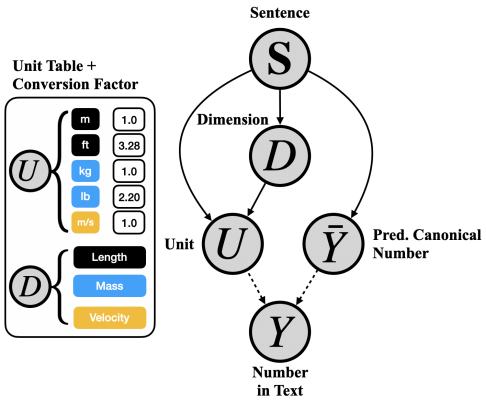


Figure 2: *GeMM* as a graphical model. The broken arrows represent a deterministic unit conversion. Examples of unit values and their corresponding dimension values are also shown.

Furthermore there are numerous applications of better measurement prediction and unit reconstruction such as in table to text generation (Moosavi et al., 2021), answering numerical queries (Sarawagi and Chakrabarti, 2014; Ho et al., 2019) or for improving e-commerce product search(Arici et al., 2021). We hope that Masked Measurement Prediction becomes a standard benchmarking tool from which we can gain insight how to best incorporate new numeracy modeling techniques as well as evaluate existing models.

2 Models

2.1 Background + Notation

The International System of Units (SI) defines seven *fundamental dimensions* (Length, Time, Mass, etc.) and seven corresponding *base SI units*

(meters, seconds, kilograms, etc.). The SI system is the most widely adopted measurement standard and is used internationally in domains such as commerce, finance, logistics, and science. We designate $\mathcal{D}$ to be the set of composite dimensions obtained from (and including) the *fundamental dimensions*. Values of $\mathcal{D}$ include velocity and power. We let $\mathcal{U}$ be the set of all units: the various ways to describe dimensions. For example, units of Length include meters and miles. Each training example consists of a real number y, a dimension $d \in \mathcal{D}$, a unit $u \in \mathcal{U}$, and the remainder of the sentence $\boldsymbol{S}$. In *MMP*, our task is to predict y, d, and u given only $\boldsymbol{S}$. In the next sections we describe our generative model designed for *MMP* followed by the ablations we consider.

2.2 Model

Measurements have complex semantic meanings, shaped by many standards, particular instruments, and natural world phenomena. Consider a text concerning rainfall. From a dimensional analysis perspective, the units *inches per year (in/y)* and *meters per second (m/s)* share the same dimension *velocity*. However, mentioning *in/y* usually implies that the text is discussing total rainfall in a region. Likewise, the use of *m/s* suggests that the text is examining the speed of falling rain droplets. To capture this complexity, we consider a generative model that learns the joint distribution of the number, dimension, and unit.

We now describe the generative process of our full model. To start, conditioned on $\boldsymbol{S}$, our model samples a discrete dimension variable $\boldsymbol{D}$. Then conditioned on the sampled dimension, our model samples a discrete unit variable $\boldsymbol{U}$ compatible with the dimension. For example, conditioned on the dimension *velocity* our model will output a distribution over the units of velocity such as *[miles per hour; meters per second, inches per year]* as opposed to all of $\mathcal{U}$. We then separately predict a distribution on the canonicalized measurement, $\bar{\boldsymbol{Y}}$, which is the numerical quantity represented in a base canonical (metric) unit like meters. During inference time, we use the highest scoring dimension and unit and choose the proper conversion factor to deterministically produce the final number y represented in the predicted unit. We refer to this **Ge**nerative **M**asked **M**easurement model as *GeMM*, where the joint $p(\boldsymbol{D}, \boldsymbol{Y}, \boldsymbol{U}|\boldsymbol{S})$ is given by the following equation:

$$p(\boldsymbol{D}|\boldsymbol{S}) \times p(\boldsymbol{U}|\boldsymbol{D}, \boldsymbol{S}) \times p(\boldsymbol{Y}|\boldsymbol{S})$$

We show the graphical model of *GeMM* in Figure 2. We also consider, *GeMM* U→Y, a slight variant where we have a direct dependence between the unit and number prediction with a joint equal to:

$$p(\boldsymbol{D}|\boldsymbol{S}) \times p(\boldsymbol{U}|\boldsymbol{D}, \boldsymbol{S}) \times p(\boldsymbol{Y}|\boldsymbol{U}, \boldsymbol{S})$$

2.3 Discrete Latent Dimension Model

We also consider an unsupervised generative model which treats the dimension as a discrete latent variable. We use the same number of dimension classes $|\mathcal{D}|$ and train to maximize the log-likelihood of the observed $\boldsymbol{Y}$. We refer to this model as *Lat-Dim* and is characterized by:

$$p(\boldsymbol{Y}|\boldsymbol{S}) = \sum_{\boldsymbol{D}} p(\boldsymbol{D}|\boldsymbol{S}) \times p(\boldsymbol{Y}|\boldsymbol{D}, \boldsymbol{S})$$

To evaluate this model we build a contingency matrix of the predicted classes and using a linear solver find the best mapping between our predicted and true dimensions. We can then apply this mapping to the model predictions and calculate classification metrics for dimension prediction.

2.4 Model Ablations

We also consider several model ablations of *GeMM*. Our first ablation is *GeMM* -Y -U which models $p(\boldsymbol{D}|\boldsymbol{S})$. The second, *GeMM* -Y, learns the distribution $p(\boldsymbol{U}, \boldsymbol{D}|\boldsymbol{S}) = p(\boldsymbol{D}|\boldsymbol{S}) \times p(\boldsymbol{U}|\boldsymbol{D}, \boldsymbol{S})$. The third, *GeMM* -U, models $p(\boldsymbol{Y}, \boldsymbol{D}|\boldsymbol{S}) = p(\boldsymbol{D}|\boldsymbol{S}) \times p(\boldsymbol{Y}|\boldsymbol{D}, \boldsymbol{S})$. Our final ablation is *GeMM* -U -D which learns $P(\boldsymbol{Y}|\boldsymbol{S})$ directly.

2.5 Model Architectures

For our textual encoder, we use the Huggingface Transformers (Wolf et al., 2020; Liu et al., 2019) implementation of RoBERTa, a pretrained 12-layer transformer. We refer to this text encoder as $\boldsymbol{T}$ such that given a sentence $\boldsymbol{S}$, our model outputs a 768-dimensional vector $\boldsymbol{h_T}$. We use a single linear layer, $W_S \in \mathbb{R}^{768 \times M}$, to project $\boldsymbol{h_T}$ to $\boldsymbol{h}$ and treat the dimension M as a hyper-parameter. To form a distribution over the real number line $\mathbb{R}$ we use a *Log-Laplace* model, a competitive model used in the numeracy literature (Spokoyny and Berg-Kirkpatrick, 2020; Thawani et al., 2021a; Zhang et al., 2020). This is equivalent to L_1 regression in log-space and yields the following loss function

where $\boldsymbol{Y}$ and $\boldsymbol{Y}^*$ are predicted and ground truth numbers, respectively:

$$\log P(\boldsymbol{Y}|\boldsymbol{S}) = |\log \boldsymbol{Y}^* - \log \boldsymbol{Y}| + \log \left| \frac{1}{\boldsymbol{Y}} \right| \quad (1)$$

As shown in Figure 1, we project $\boldsymbol{h}$ with a linear layer $W_{\boldsymbol{D}} \in \mathbb{R}^{M \times |\mathcal{D}|}$ to obtain a distribution over $\boldsymbol{D}$. We then use a separate linear layer, $W_{\boldsymbol{U}} \in \mathbb{R}^{M \times |\mathcal{U}|}$, to project $\boldsymbol{h}$ and obtain a distribution over $\boldsymbol{U}$. To predict $\bar{\boldsymbol{Y}}$, we project $\boldsymbol{h}$ with a linear layer $W_{\boldsymbol{Y}}$. In the case of *GeMM*, we let $W_{\boldsymbol{Y}} \in \mathbb{R}^{M \times |\mathcal{D}|}$ in order to parameterize a mean of a *Log-Laplace* distribution for each dimension in $\boldsymbol{D}$. For *GeMM* U→Y, we set $W_{\boldsymbol{Y}} \in \mathbb{R}^{M \times |\mathcal{U}|}$ to output the mean of a *Log-Laplace* distribution for each unit in $\boldsymbol{U}$ and the remaining models, we set $W_{\boldsymbol{Y}} \in \mathbb{R}^{M \times 1}$ resulting in a single mean of a *Log-Laplace* distribution. For training, we use cross-entropy loss for the dimension and unit distributions, and the loss from the equation above for number prediction.

3 Dataset

We train and evaluate our models on *WiCo* (Thawani et al., 2021a), a dataset of English Wikipedia sentences where the number and unit in each sentence are human-annotated. We canonicalize the units and map each to a single dimension. For example both *feet per second* and *miles per hour* map to *velocity*. We show the distribution of all measurements and *lengths* in Figure 3. The resulting dataset consists of 919,237 sentences with annotated (number, unit, dimension) triples. We provide more details on the data in Appendix A.

4 Experiments

We train all models using a batch size of 200 and use the AdamW (Loshchilov and Hutter, 2019) optimizer with a learning rate of $1e^{-4}$ and a linear warm-up schedule of 500 steps. We use the "❄" symbol to indicate that we freeze the transformer parameters for training. For all frozen models we use a log frequency weighted cross-entropy due to the highly imbalanced classes as well as a higher learning rate of $1e^{-3}$. We employ early stopping with a patience of five epochs on validation score.

To evaluate the performance of our models, we report the macro averaged F1 score for dimension and unit prediction and *log-mae* to evaluate number prediction. We define *log-mae* in Equation 2 where $\boldsymbol{Y}$ is the predicted number and $\boldsymbol{Y}^*$ is the ground truth number. As a simple baseline for dimension and unit prediction, we employ majority class voting. For number prediction we use the median of all the numbers in the training set.

Model	10-shot	40-shot	70-shot	100-shot
GeMM -Y -U ❄	15.5	50.0	52.5	53.4
GeMM -Y -U	**42.5**	**51.2**	**57.6**	**60.5**
Majority	14.3	14.3	14.3	14.3

Table 1: Results (measured by F1 ↑) of our few-shot experiment on dimension classification (probing $p(D|S)$). x-shot implies the model is trained on x labeled examples per dimension. *GeMM* -Y -U indicates an ablation of *GeMM* where $\boldsymbol{Y}$ and $\boldsymbol{U}$ are not modeled. ❄ indicates the model's parameters are frozen during training.

Model	10-shot	40-shot	70-shot	100-shot
GeMM -U -D ❄	1.94	1.82	1.72	1.75
GeMM -U -D	**1.70**	**1.56**	**1.43**	**1.41**
Median	1.99	1.99	1.99	1.99

Table 2: Results (*log-mae* ↓) of our few-shot experiment on number prediction (probing $p(Y|S)$).

$$\textit{log-mae} = \frac{1}{|\mathcal{D}_{\text{test}}|} \sum_{\mathcal{D}_{\text{test}}} |\log_{10} \boldsymbol{Y}^* - \log_{10} \boldsymbol{Y}| \quad (2)$$

4.1 Few-Shot

To study the degree to which current pretrained models capture different aspects of numeracy, we consider the following few-shot experiment. We sample a balanced dataset of dimensions where each class gets 10, 40, 70, or 100 labeled examples. We train *GeMM* -Y -U and *GeMM* -U -D on the few-shot task where the pretrained text encoder $\boldsymbol{T}$ parameters are frozen and compare their performance against full fine-tuning. Due to the high variance of *GeMM* -Y -U, we report the average of three random seeds. In Table 1 and Table 2 we show results of *GeMM* -Y -U and *GeMM* -U -D respectively.

Although performance improves with more data, the frozen models significantly underperform their unfrozen counterparts across all dataset sizes. For example, in the 100-shot dataset, the frozen model shows 7.1 lower F1 and 0.34 higher *log-mae*. These results suggest that current pretrained transformers do not capture numeracy to a large extent.

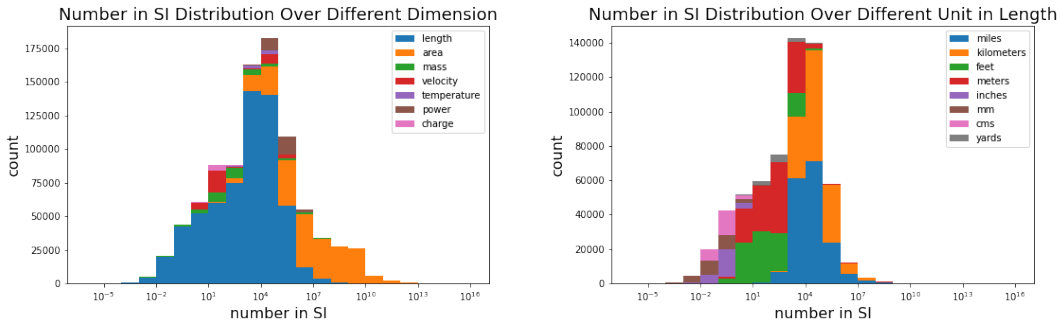


Figure 3: Histograms of *WiCo* numbers binned by base-10 exponent. All numbers are canonicalized to their SI form. **Left:** All numbers labeled by dimension. **Right:** Numbers in the *length* dimension labeled by unit.

Model	Probing Type	Val	Test
Majority	-	33.1	33.1
GeMM ❄	$p(\boldsymbol{D}\|\boldsymbol{S})$	69.1	67.5
GeMM -Y -U	$p(\boldsymbol{D}\|\boldsymbol{S})$	88.0	86.8
GeMM -Y	$p(\boldsymbol{D}\|\boldsymbol{S})$	87.0	**87.3**
GeMM -U	$p(\boldsymbol{D}\|\boldsymbol{S})$	87.2	86.6
Lat-Dim	$p(\boldsymbol{D}\|\boldsymbol{S})$	9.0	9.1
GeMM	$p(\boldsymbol{D}\|\boldsymbol{S})$	87.4	87.0
GeMM U→Y	$p(\boldsymbol{D}\|\boldsymbol{S})$	86.4	86.1

Table 3: Results (**F1** ↑) for dimension prediction conditioned on $\boldsymbol{S}$ only. *GeMM* U→Y indicates a variant of *GeMM* where $\bar{\boldsymbol{Y}}$ is dependent on $\boldsymbol{U}$ (in addition to $\boldsymbol{S}$).

Model	Probing Type	Val	Test
GeMM -U	$p(\boldsymbol{D}\|\bar{\boldsymbol{Y}}, \boldsymbol{S})$	95.5	95.7
GeMM U→Y	$p(\boldsymbol{D}\|\bar{\boldsymbol{Y}}, \boldsymbol{S})$	96.4	**96.6**

Table 4: Results (**F1** ↑) for dimension prediction conditioned on $\bar{\boldsymbol{Y}}$ and $\boldsymbol{S}$.

4.2 Dimension Prediction

We train our models and their ablations on the full dataset and measure their performance on dimension prediction. In Table 3, we show the results of dimension prediction conditioned on $\boldsymbol{S}$. We observe that the performance gap between the frozen and unfrozen *GeMM* grows to 19.5 F1 on the test split despite training on 3 orders of magnitude more training data than the few-shot setting.

By using Bayes' rule, we perform dimension prediction conditioned on both $\boldsymbol{S}$ and $\bar{\boldsymbol{Y}}$ and show our results in Table 4. We observe that both models show improved dimension prediction ability when supplied with the number with *GeMM* U→Y reaching 96.6 F1 score, an effective error rate reduction of 75%.

Model	Probing Type	Val	Test
Majority	-	8.9	9.0
GeMM ❄	$p(\boldsymbol{U}\|\boldsymbol{D}, \boldsymbol{S})$	29.8	29.8
GeMM -Y	$p(\boldsymbol{U}\|\boldsymbol{D}, \boldsymbol{S})$	52.9	51.7
GeMM	$p(\boldsymbol{U}\|\boldsymbol{D}, \boldsymbol{S})$	51.5	**54.9**
GeMM U→Y	$p(\boldsymbol{U}\|\boldsymbol{D}, \boldsymbol{S})$	49.3	47.8

Table 5: Results (**F1** ↑) on unit prediction conditioned on the true dimension and text. Ablations are above the double horizontal line.

4.3 Unit Prediction

We show the unit prediction performance of our models in Table 5. The strongest performing model for unit prediction was *GeMM* with a F1 score of 54.9. Again, the frozen *GeMM* ❄ produced a 25.1 lower F1 score than its unfrozen counterpart.

We note that even though the F1 scores on unit prediction are much lower than dimension prediction, they are still significantly better than the majority baseline. Although one can freely substitute a unit with one in the same dimensional class, we tend to be more systematic and choose units that allow for more straightforward human readability or reflect the actual instruments used for measurement. As a result, we gravitate towards regularities that models can learn to recognize. The converse of this is also interesting as it suggests that the expressed units imply more semantic meaning than what is captured in the standardized measurement.

Model	Probing Type	Val	Test
Median	-	1.98	1.97
GeMM ❄	$p(\bar{\boldsymbol{Y}}\|\boldsymbol{S})$	1.377	1.370
GeMM -U -D	$p(\bar{\boldsymbol{Y}}\|\boldsymbol{S})$	0.529	0.531
GeMM -U	$p(\bar{\boldsymbol{Y}}\|\boldsymbol{D},\boldsymbol{S})$	0.468	0.469
	$p(\bar{\boldsymbol{Y}},\boldsymbol{D}\|\boldsymbol{S})$	0.517	0.518
Lat-Dim	$p(\bar{\boldsymbol{Y}},\boldsymbol{D}\|\boldsymbol{S})$	0.545	0.546
GeMM	$p(\bar{\boldsymbol{Y}}\|\boldsymbol{S})$	0.517	**0.515**
GeMM U→Y	$p(\bar{\boldsymbol{Y}}\|\boldsymbol{U},\boldsymbol{D},\boldsymbol{S})$	0.401	**0.401**
	$p(\bar{\boldsymbol{Y}},\boldsymbol{U},\boldsymbol{D}\|\boldsymbol{S})$	0.526	0.526

Table 6: Results (*log-mae* ↓) for number prediction conditioned on $\boldsymbol{S}$. In the second row of *GeMM* -U, we select the highest scoring $d^* \in \boldsymbol{D}$ and predict y conditioned on d^* and $\boldsymbol{S}$. In the second row of *GeMM* U→Y, we select the highest scoring $u^* \in \boldsymbol{U}$ and $d^* \in \boldsymbol{D}$ and predict y conditioned on u^*, d^*, and $\boldsymbol{S}$. For *Lat-Dim*, we sum over the latent variable $\boldsymbol{D}$ to predict y conditioned on $\boldsymbol{S}$.

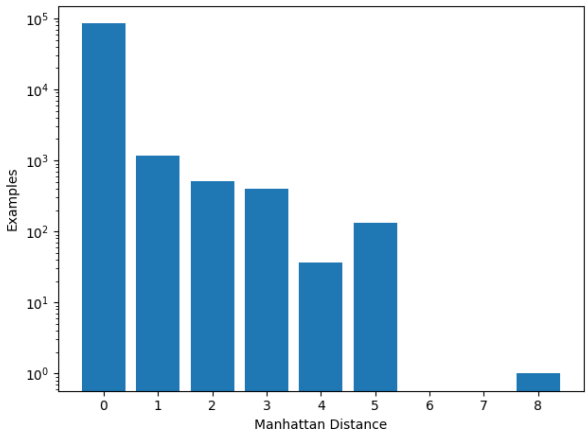


Figure 4: Manhattan distance between true and predicted dimensions by *GeMM* U→Y. We treat dimensions as vectors whose elements are the exponents of the fundamental dimensions that compose a given dimension. Note that the y-axis is in log-scale.

4.4 Number Prediction

We show the number prediction performance of our models in Table 6. Consistent with our previous experiments, all models outperform *GeMM* ❄. Furthermore, we observe that not modeling $\boldsymbol{U}$ and $\boldsymbol{D}$ (as is the case in *GeMM* -U -D) increases *log-mae*, i.e., results in worse numerical prediction. While competitive with *GeMM* and its variants on number prediction, *Lat-Dim* cannot predict dimensions with the same efficacy (Table 3).

We also experiment with the setting where *GeMM* -U conditionally generates the number for a particular dimension. In this setting, *GeMM* -U improves *log-mae* to 0.469. Extending this setting further, we condition *GeMM* U→Y on both a unit and a dimension to produce the best *log-mae* among our models: 0.401.

We now revisit our original motivating example: *"Alex Honnold climbed for [NUM] [UNIT]"*. Assume we want to know the distance of a climb. To do this, we condition *GeMM* U→Y on $\boldsymbol{D} = length$ and $\boldsymbol{U} = feet$. If, on the other hand, we want to know the duration of a climb, we change the conditioning to $\boldsymbol{D} = time$ and $\boldsymbol{U} = hours$. Now, if we want to know the length of Alex Honnold's climbing career, we condition *GeMM* U→Y on $\boldsymbol{D} = time$ and $\boldsymbol{U} = years$. These examples illustrate the flexibility of *GeMM* U→Y and the importance of jointly modeling numbers, units, and dimensions.

4.5 Quantitative Analysis

4.5.1 Dimensions and Unit

In Figure 7a shown in Appendix C.1, we visualize a confusion matrix of dimension predictions by *GeMM* U→Y. The low accuracy for electric charge and temperature is attributed to a mislabeling in the dataset.[3] For mass, we find many ambiguous situations where either mass or length are appropriate. See the first row of Table 9 for such an example.

Thus far, we have treated dimensions as distinct classes with no relationships. However, dimensions are compositions of the seven fundamental dimensions. Therefore, dimensions that share fundamental dimensions are more similar than those that do not. To quantify this similarity, we can treat dimensions as a vector where each element represents the exponent of a fundamental dimension. Then to measure the similarity of two dimensions, we take their Manhattan distance. To illustrate, assume there exist only two fundamental dimensions: Length and Time. Let $speed = (1, -1)$ and $length = (1, 0)$ where the first element represents Length and the second represents Time. The Manhattan distance between $speed$ and $length$ is equal to one. In Figure 4, we visualize the Manhattan distance between the predictions of *GeMM* U→Y and ground truth. We observe that there is generally an inverse relationship between error count and the distance of the errors. This observation suggests

[3] Sentences with mislabeled `Celsius` as `Coulombs`, which may due to wrong annotation between °C and C. Also observed by Elazar et al. (2019)

Length	Area	Velocity	Mass	Power
0.37	0.54	0.19	0.55	0.27

Table 7: *log-mae* ↓ by dimension. It is harder to predict numbers of Area and Mass than other dimensions.

	Model		Human		Model > Human
	D	***U***	***D***	***U***	***Y***
Tech Ann.	**96.7**	**86.2**	88.9	46.3	**78.8**
AMT Ann.	**96.7**	**77.0**	89.3	43.5	**66.2**

Table 8: Dimension and unit prediction accuracy of our human evaluation experiment. *GeMM* U→Y outperformed the human annotators in both evaluations. **Tech Ann.** is over a balanced set of 90 sentences labeled by Technical Annotators. **AMT Ann.** is over a balanced set of 2,122 sentences annotated by AMT Annotators. The final column shows the model predicted a number closer to ground truth in 66.2-78.8% of the cases.

that our model has learned that some dimensions are more similar than others. This suggestion is reinforced by Figure 7a where misclassifications tend to have small distances from the true dimension. For example, velocity is most often misclassified as length. For unit prediction, we find that most mistakes occur substituting units with ones that have similar magnitudes like feet for meters or kilometers for miles (see Appendix 7b).

4.5.2 Numeracy

In Table 7, we show *log-mae* by dimension as predicted by *GeMM* U→Y. We note that errors are not uniform across dimensions, predicting *areas* is 2.2 times harder *velocities*. We also observe that the magnitudes of errors seem to be positively correlated with the variances observed in Figure 3.

4.5.3 Human Evaluation

We perform two evaluations of *GeMM* U→Y against human annotators. In the first evaluation, we compare against the combined effort of three Technical Annotators on a balanced set of 90 sentences randomly sampled from the test set. The annotators worked together to predict the missing dimensions, units, and accurate measurement estimates. Examples of sentences and annotations shown in Table 9.

In the second evaluation, we compare against Amazon Mechanical Turk (AMT) Annotators on a balanced set of 2,122 sentences randomly sampled from the test set. We show the results for both evaluations in Table 8.

In both evaluations, the model outperforms the human annotators on every task. For dimension prediction, the model led by 7.4-7.8 percentage points. Of the sentences where the dimension was correctly annotated, the model led by 33.5-39.9 percentage points on unit prediction. For sentences where both the model and human correctly predicted the dimension, the model predicted a number closer to ground truth 66.2-78.8% of the time.

4.6 Qualitative Analysis

4.6.1 Semantic Head Embeddings

In Figure 5 we plot the t-SNE embeddings of the sentences' $\boldsymbol{h}$, the output of our text encoder. We label each $\boldsymbol{h}$ with the masked measurement's true dimension, unit and exponent of the number. In 5a we observe that most embeddings labeled by their true dimension tend to form tight clusters. In 5b we filter to only show embeddings that share the *Length* dimension and label them by their units. We find that clusters are organized by the relative magnitudes of their units: large (*Kilometers, miles*), medium (*feet, meters*), and small (*millimeters, inches, centimeters*). Further we see that *yards* appear close to other *imperial units* of *feet* and *miles*. Finally, in 5c when embeddings are binned by the exponent of their values we observe that the left to right direction appears to capture the increasing magnitude of a number.

5 Related Work

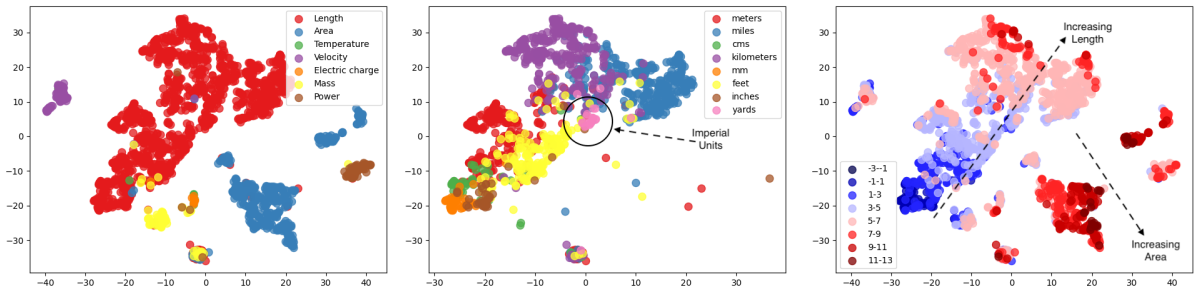
5.1 Numeracy

Multiple works have probed word embeddings like word2vec, GloVe, FastText (Naik et al., 2019) and contextual embeddings from models like BERT (Wallace et al., 2019; Zhang et al., 2020) or T5 (Pal and Baral, 2021) on a variety of numerical tasks like sorting, numeration, magnitude prediction, and common sense (Lin et al., 2020). Several works have targeted numeracy pretraining using left to right language models (Spithourakis and Riedel, 2018), CNN and RNN based models (Chen et al., 2019), pretrained transformers (Spokoyny and Berg-Kirkpatrick, 2020; Jin et al., 2021), for an overview (Thawani et al., 2021b).

Incorporating synthetic mathematical data augmentations (Geva et al., 2020) has improved question answering while numerical pretraining has been shown to lower masked language modelling

#	Text	True			*GeMM* U-Y Prediction			Human Prediction		
		Dim	Unit	Num	Dim	Unit	Num	Dim	Unit	Num
1	Hope is gaff rigged, 'V'-bottomed and has an [#NUM] [UNIT] centerboard.	Mass	pounds	385.6	Length	feet	2.97	Length	meter	50
2	Some have been running for over 50 years, each covering about [#NUM] [UNIT].	Velocity	miles/year	0.10	Area	sqkm	2.09E+10	Area	sqmi	2.59E+07
3	Another medium-sized corvid, the [#NUM] [UNIT] Eurasian magpie (Pica pica) is also amongst the most widely reported secondary prey species for goshawks there.	Mass	grams	0.22	**Mass**	**grams**	0.05	**Mass**	**grams**	**0.2**
4	The twin cylinder, liquid-cooled, in-line two-stroke, [#NUM] [UNIT] Rotax 582 has also been used.	Power	horse-power	47725	**Power**	**horse-power**	39248	**Power**	**horse-power**	**45000**
5	Chrysothamnus may grow up to a [#NUM] [UNIT] tall shrub or subshrub, usually with woody stem bases	Length	cms	1.2	**Length**	meters	**1.147**	**Length**	meters	1
6	Kurt Busch was the fastest in the first practice session with a time of 21.372 seconds and a speed of [#NUM] [UNIT].	Velocity	miles/hour	75.1	**Velocity**	**miles/hour**	**63.584**	**Velocity**	meters/second	10

Table 9: Instances of the *MMP* task performed during our human evaluation experiment, all numbers are in SI units. In ex. 1, both the model and humans predict the incorrect dimension length instead of mass. The preceding sentence of ex. 2 references "trains" leading both to incorrectly predict area instead of velocity. In ex. 6 the model predicts the speed of the NASCAR driver Kurt Busch's car whereas the humans had mistaken him for a runner.



(a) (b) (c)

Figure 5: t-SNE visualizations of semantic head embeddings labeled by **(left 5a)** dimension, **(middle 5b)** units of *length*, and **(right 5c)** number exponent bin. **Middle**: we observe a clustering of imperial units: feet, yards, miles. **Right**: we show two directions where magnitudes of length and area measurements increase in value.

perplexity (Thawani et al., 2021a). Either directly or indirectly units have been involved in providing more interpretable explanation of quantities (Chaganty and Liang, 2016), solving Fermi problems (Kalyan et al., 2021) and resolving numeric Fused-Heads (Elazar and Goldberg, 2019).

5.1.1 Numeracy Benchmarks

Several numeracy benchmarks have been proposed like quantitative reasoning in natural language entailment (Ravichander et al., 2019) and synthetic measurement estimation (Jin et al., 2021). The closest benchmark to our work is the Distribution over Quantities dataset (DoQ) introduced by Elazar et al. (2019). A rule-based method was combined with simple heuristics to build DoQ resulting in its high-coverage albeit also higher noise. Although, *WiCo* is smaller, it has much higher fidelity since it utilizes a feature used by editors of Wikipedia to automatically convert quantities into different units. Further, *WiCo* provides the whole sentence as context as opposed to triplets of words. Zhang et al. (2020) use artificial templates to probe models on DoQ and find little difference between numerically pretrained and frozen embeddings such as ELMo. In contrast, our findings show there is a significant gap on *WiCo* between fully finetuned models and their frozen counterparts.

6 Conclusion

In this work we propose Masked Measurement Prediction, a new task to resolve the limitation of masked number prediction in which units are not considered. In our study, we show probing of traditional pretrained transformers exposes a gap in their understanding of contextualized quantities. Through careful quantitative and qualitative analysis of our new model, which directly reasons about underlying units and dimensions, we find that it is possible to learn good representations of measurements. For future work we aim to extend this dataset to cover more existing standardized units from organizations such as UNECE.[4] We hope our *MMP* task encourages research into further development of better numeracy methodologies.

[4] United Nations Economic Commission for Europe

References

Tarik Arici, Kushal Kumar, Hayreddin Çeker, K K Saladi, and Ismail B. Tutar. 2021. Solving price per unit problem around the world: Formulating fact extraction as question answering. In *KDD TrueFact Workshop*.

Arun Chaganty and Percy Liang. 2016. How much is 131 million dollars? putting numbers in perspective with compositional descriptions. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 578–587, Berlin, Germany. Association for Computational Linguistics.

Chung-Chi Chen, Hen-Hsen Huang, Hiroya Takamura, and Hsin-Hsi Chen. 2019. Numeracy-600K: Learning numeracy for detecting exaggerated information in market comments. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6307–6313, Florence, Italy. Association for Computational Linguistics.

Wenhu Chen, Hongmin Wang, Jianshu Chen, Yunkai Zhang, Hong Wang, Shiyang Li, Xiyou Zhou, and William Yang Wang. 2020. Tabfact : A large-scale dataset for table-based fact verification. In *International Conference on Learning Representations (ICLR)*, Addis Ababa, Ethiopia.

Zhiyu Chen, Wenhu Chen, Charese Smiley, Sameena Shah, Iana Borova, Dylan Langdon, Reema Moussa, Matthew I. Beane, Ting-Hao Kenneth Huang, Bryan R. Routledge, and William Yang Wang. 2021. Finqa: A dataset of numerical reasoning over financial data. In *EMNLP*.

Yanai Elazar and Yoav Goldberg. 2019. Where's my head? definition, data set, and models for numeric fused-head identification and resolution. *Transactions of the Association for Computational Linguistics*, 7:519–535.

Yanai Elazar, A. Mahabal, Deepak Ramachandran, Tania Bedrax-Weiss, and Dan Roth. 2019. How large are lions? inducing distributions over quantitative attributes. In *ACL*.

Mor Geva, Ankit Gupta, and Jonathan Berant. 2020. Injecting numerical reasoning skills into language models. In *ACL*.

Vinh Thinh Ho, Yusra Ibrahim, Koninika Pal, Klaus Berberich, and Gerhard Weikum. 2019. Qsearch: Answering quantity queries from text. In *SEMWEB*.

Zhihua Jin, Xin Jiang, Xingbo Wang, Qun Liu, Yong Wang, Xiaozhe Ren, and Huamin Qu. 2021. Numgpt: Improving numeracy ability of generative pre-trained models. *ArXiv*, abs/2109.03137.

A. Kalyan, Abhinav Kumar, Arjun Chandrasekaran, Ashish Sabharwal, and Peter Clark. 2021. How much coffee was consumed during emnlp 2019? fermi problems: A new reasoning challenge for ai. *ArXiv*, abs/2110.14207.

Bill Yuchen Lin, Seyeon Lee, Rahul Khanna, and Xiang Ren. 2020. Birds have four legs?! numersense: Probing numerical commonsense knowledge of pre-trained language models. *ArXiv*, abs/2005.00683.

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. 2019. Decoupled weight decay regularization. In *ICLR*.

Nafise Sadat Moosavi, Andreas Rücklé, Dan Roth, and Iryna Gurevych. 2021. Learning to reason for text generation from scientific tables. *arXiv preprint arXiv:2104.08296*.

Aakanksha Naik, Abhilasha Ravichander, Carolyn Rose, and Eduard Hovy. 2019. Exploring numeracy in word embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3374–3380, Florence, Italy. Association for Computational Linguistics.

Kuntal Kumar Pal and Chitta Baral. 2021. Investigating numeracy learning ability of a text-to-text transfer model. *ArXiv*, abs/2109.04672.

Qiu Ran, Yankai Lin, Peng Li, Jie Zhou, and Zhiyuan Liu. 2019. NumNet: Machine reading comprehension with numerical reasoning. 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 2474–2484, Hong Kong, China. Association for Computational Linguistics.

Abhilasha Ravichander, Aakanksha Naik, Carolyn Rose, and Eduard Hovy. 2019. Equate: A benchmark evaluation framework for quantitative reasoning in natural language inference. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 349–361.

Subhro Roy. 2017. *Reasoning about quantities in natural language*. Ph.D. thesis, University of Illinois at Urbana-Champaign.

Mark Sammons, V.G.Vinod Vydiswaran, and Dan Roth. 2010. "ask not what textual entailment can do for you...". In *ACL*.

Sunita Sarawagi and Soumen Chakrabarti. 2014. Open-domain quantity queries on web tables: annotation, response, and consensus models. *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*.

Georgios P Spithourakis and Sebastian Riedel. 2018. Numeracy for language models: Evaluating and improving their ability to predict numbers. *arXiv preprint arXiv:1805.08154*.

Daniel Spokoyny and Taylor Berg-Kirkpatrick. 2020. An empirical investigation of contextualized number prediction. In *EMNLP*.

Avijit Thawani, Jay Pujara, and Filip Ilievski. 2021a. Numeracy enhances the literacy of language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6960–6967, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Avijit Thawani, Jay Pujara, Filip Ilievski, and Pedro Szekely. 2021b. Representing numbers in NLP: a survey and a vision. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 644–656, Online. Association for Computational Linguistics.

Eric Wallace, Yizhong Wang, Sujian Li, Sameer Singh, and Matt Gardner. 2019. Do nlp models know numbers? probing numeracy in embeddings. In *Empirical Methods in Natural Language Processing*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Xikun Zhang, Deepak Ramachandran, Ian Tenney, Yanai Elazar, and Dan Roth. 2020. Do language embeddings capture scales? In *FINDINGS*.

A Dataset

We train and evaluate our models on Wiki-Convert (*WiCo*) (Thawani et al., 2021a), a dataset of English Wikipedia sentences where the number and unit in each sentence are human-annotated. The built-in template in Wikipedia can ensure the text contains numbers and units. For example, `{{convert|2|km|mi}}` displays as `2 kilometres (1.2 mi)`. By searching within Wikipedia articles for the use of this template, the authors of *WiCo* automatically extract human-annotated numbers. To perform unit canonicalization, we use Pint [5] whenever the mapping is unambiguous. In the ambiguous case, we manually inspect the sentence and perform the mapping. For example, we map the unit `sqmi` in *WiCo* to `square miles` to let pint perform unit canonicalization. Table 9 shows examples of the extended dataset. The original dataset contains 924,473 sentence. The median sentence length is 106 characters, with 29,597 sentences has a length shorter than 20 characters. We provide statistics of the data in Table 10. For preprocessing we exclude sentences which have more than 64 tokens to have efficient computing memory or where the number is negative for simplicity. According to Thawani et al. (2021a) *WiCo*, "... has been extracted from Wikipedia dumps, which are licensed under the GNU Free Documentation License (GFDL) and the Creative Commons Attribution-Share-Alike 3.0 License." Thawani et al. (2021a) constructed *WiCo* with the intent that it be used to further numeracy NLP research. Our use of *WiCo* is aligned with its authors' goals.

[5] Pint: https://github.com/hgrecco/pint

Split	Examples	Max #	Min #
All	919,237	5.5E+36	1E-06
Train	728,629	5.5E+36	1E-06
Val	91,110	4.4E+14	1.2E-06
Test	91,092	1.6E+21	1.8E-06

Table 10: Summary statistics for Wiki-Convert. The median number of characters and tokens per example is 106 and 33, respectively.

B MLM Preliminary Unit Probe

We perform a preliminary unit probe shown in Table 11. The model predicts vastly different numbers when conditioned on different units. We observe a mean of 3086.8 and a standard deviation of 5820 for all the converted metric output.

C Experiments

We train our model *GeMM* U→Y on a single Nvidia GeForce RTX 2080 Ti for 4 hours and 14 minutes with a total parameter of 124,696,538.

C.1 Quantitative Analysis

In Figure 6, we show *log-mae* is relatively small for small magnitude units, which means predicting numbers for small magnitude units is easier than predicting numbers for their larger counterparts.

In Figure 7, we show confusion matrices of dimension and unit predictions by *GeMM* U→Y.

Input: [UNIT]	m	km	ft	mi	yd	in	meters	kilometers	feet	miles	yards	inches	-
Output	200	10	200	2	100	1	200	20	20	2	50	3	-
Convertion factor	1	1000	0.3048	1609.34	0.9144	0.0254	1	1000	0.3048	1609.34	0.9144	0.0254	-
Metric Output	200.0	10000.0	60.96	3218.68	91.44	0.0254	200.0	20000.0	6.096	3218.68	45.72	0.0762	-
Mean (Metric Output)	-												3086.8 m
std (Metric Output)	-												5820 m

Table 11: Example outputs for **Alex Honnold climbed for [MASK] [UNIT]**.

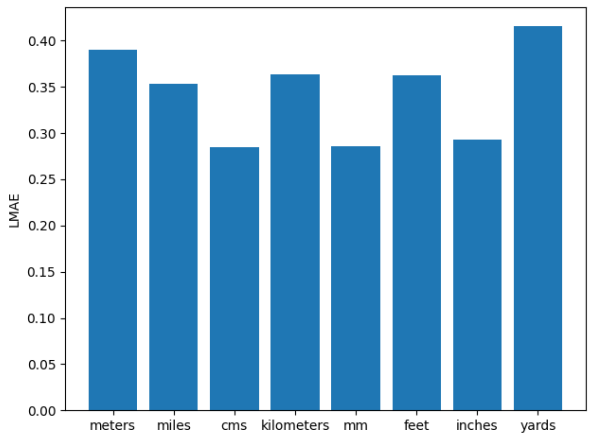


Figure 6: *log-mae* ↓ by units of length. Predicting numbers for small magnitude units is easier than predicting numbers for their larger counterparts.

D Human Annotators

D.1 Evaluation 1

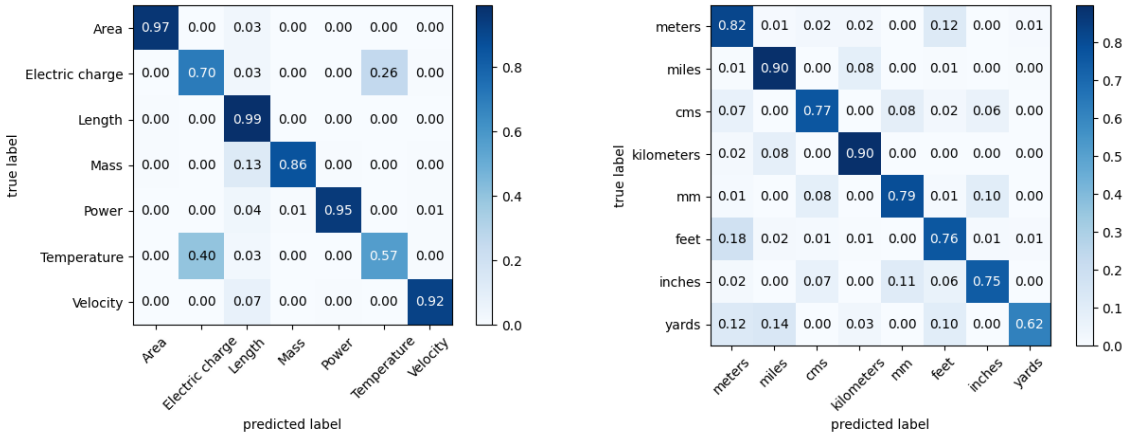
The Technical Annotators have diverse scientific backgrounds ranging from chemistry, earth sciences, and computer science. One annotator is a native Chinese speaker, and two are native English speakers.

D.2 Evaluation 2

In Figure 8 we show the instructions provided along with the interface we designed for our *MMP* task. While the workers' geographic location were not provided to us by Mechanical Turk, we aimed to compensate the workers above the US federal minimum wage of $7.25. We paid workers $0.15 per annotation with an average completion time of 33 seconds. This equates to an hourly rate of $12.80 after Mechanical Turk fees. Other demographic information is only provided by Mechanical Turk for an extra fee.

E Ethical Considerations

Like any system that makes predictions, those made by *GeMM* are not necessarily accurate and may be used by malicious actors to generate fake information to mislead their audience. Additionally, *GeMM* is an extension of RoBERTa and therefore inherits the biases learned during the training of RoBERTa. Our work focuses exclusively on English and Arabic numerals. As noted by Thawani et al. (2021a), the units in *WiCo* are heavily biased towards European and American units as they are over-represented in English Wikipedia.



(a) (b)

Figure 7: Confusion matrices for predictions by *GeMM* U→Y over the validation split. **Left 7a:** Dimension prediction. Most misclassified dimensions are similar to their ground truth counterparts in terms of Manhattan distance. **Right 7b:** Unit prediction for examples that share the *length* dimension. Most misclassified units of length share similar magnitudes to their ground truth units.

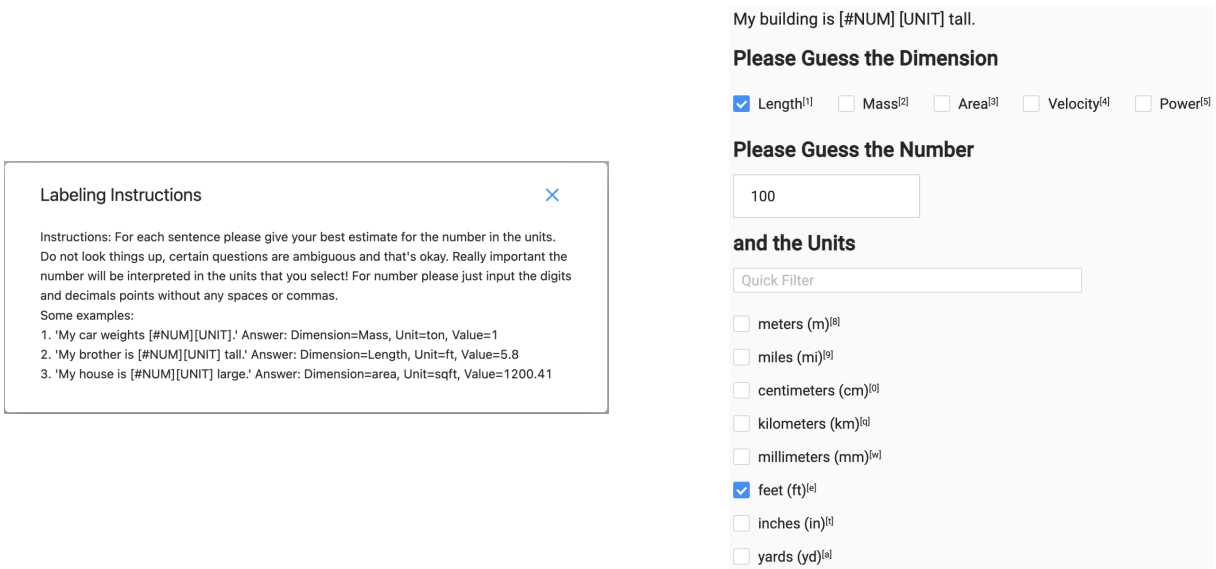


Figure 8: **Left**: Instructions for labeling task. **Right**: we show the interface used by the labelers