Stanford MLab at SemEval-2021 Task 8: 48 Hours Is All You Need

Patrick Liu, Niveditha Iyer, Erik Rozi, Ethan A. Chi
Stanford University
{pliu1, nivsiyer, erikrozi}@stanford.edu, ethanchi@cs.stanford.edu

Abstract

This paper presents our system for the Quantity span identification, Unit of measurement identification and Value modifier classification subtasks of the MeasEval 2021 task. The purpose of the Quantity span identification task was to locate spans of text that contain a count or measurement, consisting of a value, usually followed by a unit and occasionally additional modifiers. The goal of the modifier classification task was to determine whether an associated text fragment served to indicate range, tolerance, mean value, etc. of a quantity. The developed systems used pre-trained BERT models which were fine-tuned for the task at hand. We present our system, investigate how architectural decisions affected model predictions, and conduct an error analysis. Overall, our system placed 12 / 19 in the shared task and in the 2nd place for the Unit subcategory.

1 Introduction

The growing ease of access to large bodies of scientific information and research has not been accompanied by an improvement in ease of analysis or understanding of scientific text. While the performance of natural language processing tasks such as part-of-speech (POS) tagging and dependency parsing have seen improvements, analyzing counts and measurements in scientific texts has remained a largely unaddressed problem, though some work has been done by (Berrahou et al., 2013).

Measurements involve quantification (in units such as litres or kilograms), entities and the measured properties (e.g. growth rate of a fungus) and value modifiers (e.g. mean, range, tolerance). Extracting semantic relations between quantities, entities, value modifiers, and units of measurement and deriving meaning from those components is challenging because scientific communication can be ambiguous or inconsistent. In addition, the location of this information relative to the measurement tends to vary.

The precise extraction and contextualization of measurements could enable accurate summarization of large bodies of scientific literature. Additionally, this could allow for unordered measurement and numeric data from scientific texts to be transformed into standardized ordered data for easier analysis.

In this paper, we describe a rapidly implemented, lightweight model that extracts measurement context in natural language. Our pipeline was implemented in approximately 48 hours by the first three authors, who had minimal formal neural NLP experience. Despite this, we were able to achieve strong results in the two categories (*Quantity* and *Unit*) that we entered, achieving second place in the *Unit* category and 12 / 19 overall.

2 Background

2.1 MeasEval Task Setup

Data for this task is made available through the MeasEval repository (Harper et al., 2021). Input data are formatted as a paragraph of scientific text in English from which measurements would be extracted. Text annotations are provided in the BRAT Annotation format (Stenetorp et al., 2012) and evaluated in a tab separated value format.

These annotations are provided in four types of spans. A **Quantity** contains a measurement of an entity, such as *300 ml* or *10%*. **Quantities** are associated with their respective *Unit*, such as *ml* or % respectively. Each **Quantity** can also contain additional **Modifiers** which will be outlined later. A **MeasuredEntity** contains the entity the **Quantity** is referring to, such as a *flask* or other object. Subtasks 3-5 also include identifying **MeasuredEntities**, **MeasuredProperties** and **Qualifiers**, though due to time constraints, this paper mainly focuses

Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021), pages 1245–1248
Bangkok, Thailand (online), August 5–6, 2021.

on identifying **Quantities** and **Units** (subtasks 1 and 2).

For example, given the sentence *"However, some elements are only present for the largest bed inventory of 13 kg e.g., Ti, Cr, and Mn."*, an identifiable **Quantity** is *13 kg*, with the **Unit** being *kg*. Each paragraph can have multiple quantities, modifiers and units associated with them.

2.2 Prior Work

The system this paper outlines is based on the BERT architecture (Devlin et al., 2019). BERT, which stands for Bidirectional Encoder Representations from Transformers, employs the self-attention based Transformer mechanism described in (Vaswani et al., 2017). BERT allows for pre-trained bidirectional representations to be used for each word token, which can then be applied to a wide variety of tasks. As such, only a limited amount of fine-tuning is required to model NLP tasks including sentence classification and sequence tagging.

One NLP task that is especially relevant for the Unit of Measurement identification task is sequence tagging and Named Entity Recognition (NER). This task involves identifying a span of text (such as a person or location) from a given body of text. There has been significant research in the past dedicated towards NER (Li et al., 2018). (Devlin et al., 2019) also addresses sequence tagging directly in their original paper describing BERT. Therefore, we found that applying BERT towards the shared task would prove transferable and effective.

3 System Overview

3.1 Quantity Identification

Since subtask 1, quantity span identification, involves the classification of individual phrases, we fine-tuned a pre-trained BERT-Large model (Devlin et al., 2019) to perform token-level classification on an IO labeling strategy based on the IOB labeling scheme (Ramshaw and Marcus, 1995). Each token in the training and trial sets is labeled as "I" if it is inside a labeled quantity, and "O" if it is outside the labeled quantity. Since consecutive entities are very rare, we do not use the "B" label. Such an encoding scheme is common for named-entity recognition. Tokenization is implemented using BertTokenizerFast from the Hugging Face Transformers library. [1]

As with the tokenizer, we used Hugging Face library for a pre-trained BERT model. Instead of using the base BERT hidden layers (BERTModel) and manually building a few fine-tuning layers for classification, we decided to use BERT's built-in BERTForTokenClassification class, which adds a single linear layer for classification on top of the hidden layers. This choice was made in the interest of saving programming time; we did not believe that adding multiple fine-tuning layers would cause a significant increase in accuracy to justify the time that would be spent. The model outputs labels for each of the tokens, labeling them with the aforementioned IO labeling scheme. Sequences of consecutive "I" labels are then recombined into segments of text, which we use as our predicted quantities.

We trained this model on a cross entropy loss metric, with loss on "I" labels upweighted by a factor of K to account for the low relative number of quantity tokens versus non-quantity tokens. We optimized on minibatches using the Adam optimizer (Kingma and Ba, 2015), and applied a cosine-annealing scheduler (Loshchilov and Hutter, 2016) to scale the down learning rate. Both of these have been shown to be effective in fine-tuning pretrained Transformer language models.

3.2 Unit Identification

Once a quantity had been identified, the next step involved identifying the respective unit for each quantity. For consistency, we once again used a BERT tokenizer to achieve this task. Intrinsically, unit identification is considered a NER task; therefore, by using Hugging Face Transformers and the BERTForTokenClassification class, we implemented a quick, yet robust method for identifying units.

The Unit Identification approach is essentially the same as the quantity identification approach: the model labels tokens as either "I" or "O" based on whether each token is considered to be inside or outside a quantity. This approach proved to be ubiquitous and simple to implement. We identified that tokens for units were consecutive, so this model was tasked with searching for a specific word or words that could be identified as units. For example, given the Quantity *20%*, the % token would

[1] https://huggingface.co/transformers/model_doc/bert.html

be labeled as an "I" while all other tokens would be labeled as "O". By implementing a consistent approach, we tested the capabilities of pretrained robust models such as BERT, which proves to be useful for creating robust models with low computational resources.

3.3 Modifier Multilabel Classification

In the final part of subtask 2, we were tasked to classify Quantity spans by "Modifiers." Unlike the previous two parts of the pipeline, we approached this task as a multilabel classification task. Each quantity could be associated with multiple labels, such as *IsRange* and *IsApproximate*, or no labels at all. This model mapped out each label as an individual class, such that there were 10 total classes.

To predict the likelihood that a Quantity span is associated with a class, we first tokenized each quantity using BERT. These tokens were then passed into a simple one layer dropout and linear model which output 10 features. From here, each feature was considered as an independent binary classification problem in order to predict multiple classes. For simplicity, any logit greater than 0.5 was predicted as a "true" value. Binary Cross Entropy loss was used as the criterion for training.

This is a simple, yet effective approach. As this approach uses the same BERT models as the first two sections, with the key difference being the classification problem itself, we demonstrate the versatility of BERT for these tasks. Once again, we used the Hugging Face Transformers library for a pre-trained BERT model, which allowed for quick deployment. The results obtained in later sections highlight how quick and inexpensive models can still obtain acceptable results.

4 Experimental Setup

For all of the subtasks, we used the provided train-trial split of 2531 annotations across 249 texts versus 832 annotations across 66 texts.

Besides tokenizing the text and applying IO labels (see section 3.1) to get the data into a format fit for the BERT token classifier, we did not preprocess the data significantly. We tested filtering unrecognizable and irrelevant characters using regex but found no meaningful improvement in performance. We thus decided to omit preprocessing to keep with the simplistic theme of this paper.

During the hyperparameter tuning phase, we compared models based on token-level F1 score, implemented using the classification report from the metrics module of PyTorch, yielding results consistent with the provided testing script, which evaluated on the F1 overlap score. Due to the limit on the total number of iterations we could go through due to the deadline, we tuned hyperparameters by hand. We found that out of the hyperparameters we tuned, the scheduler hyperparameters were irrelevant because the scheduler was rarely used at all. The number of training epochs likewise did not come into play so long as we let the training continue until trial loss stopped decreasing; around 15 epochs was almost always enough. The only hyperparameters that seemed to significantly impact training results were the learning rate and the weight difference between quantity and non-quantity labels. After some manual tuning, we found that a learning rate of around 10^{-5} and no weight-difference adjustment performed best on the trial set for this task, resulting in our final token-level quantity identification F1 score of 0.85 on the trial set.

For the quantity identification subtask, several post-processing steps were implemented to correct simple errors in our model's predictions, based on common errors detected via manual error analysis in the model's predictions. First, commas and spaces were removed from the ends of the predicted quantities to correct for potential small character offsets on the edges of predictions, from either model error or the tokenization-detokenization process. Furthermore, all quantities that did not contain either numeric characters or strings that commonly occur in numbers (such as *two* or *-teen*) were discarded from the prediction pool. Additionally, all prepositions from a hand-picked set like *beyond* and *at* were cut from the front of each input string in order to better isolate numeric quantities from relational terms. The trial set results with and without post-processing are shown in Table 1. All the post-processing steps reduce the amount and length of predicted quantity spans. Therefore, the significantly higher precision score after post-processing indicates that the base model tends to overpredict identified quantities.

5 Results

Our model's overall precision, recall, and F1 score on overlap on the trial set are as shown in Table 1. The model's performance on evaluation metrics across the subtasks are shown in Table 2.

Metric	With	Without
Precision	0.8153	0.564
Recall	0.2912	0.301
F1 (overlap) score	0.2615	0.211

Table 1: Trial set results with and without post-processing

Category	F1 score	Ranking
Overall	0.272	12
Quantity	0.818	10
Unit	0.76	2
Modifier	0.408	8

Table 2: Evaluation set results and ranking (out of 19)

These results are promising given the speed of implementation of our system. Across all 5 subtasks, our final F1 score was 0.272. Overall, the model ranked 12th out of 19 teams in the evaluation phase of MeasEval: Counts and Measurements.

6 Conclusion

We propose a BERT-based model for information extraction of measurements and their contextual qualifiers from text. As scientific texts become increasingly open for public consumption, we hope that such systems will help present findings to broader audiences in an accessible manner. Going forward, possible areas of exploration include replacing BERT with more powerful models and augmenting training data that the model frequently makes erroneous predictions on.

Acknowledgments

This research effort would not have been possible without the support of Stanford ACM Lab. The authors thank Corey Harper, Jessica Cox, Ron Daniel, Paul Groth, Curt Kohler and Antony Scerri for organising SemEval 2021 Task 8: MeasEval - Counts and Measurements. We also thank Jillian Tang for helpful discussions.

References

Soumia Lilia Berrahou, Patrice Buche, Juliette Dibie-Barthélemy, and Mathieu Roche. 2013. How to extract unit of measure in scientific documents? In *KDIR/KMIS 2013 - Proceedings of the International Conference on Knowledge Discovery and Information Retrieval and the International Conference on Knowledge Management and Information Sharing, Vilamoura, Algarve, Portugal, 19 - 22 September, 2013*, pages 249–256. SciTePress.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Corey Harper, Jessica Cox, Curt Kohler, Antony Scerri, Ron Daniel Jr., and Paul Groth. 2021. SemEval 2021 task 8: MeasEval – extracting counts and measurements and their related contexts. In *Proceedings of the Fifteenth Workshop on Semantic Evaluation (SemEval-2021)*, Bangkok, Thailand (online). Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Jing Li, Aixin Sun, Jianglei Han, and Chenliang Li. 2018. A survey on deep learning for named entity recognition. *CoRR*, abs/1812.09449.

Ilya Loshchilov and Frank Hutter. 2016. SGDR: stochastic gradient descent with restarts. *CoRR*, abs/1608.03983.

Lance A. Ramshaw and Mitchell P. Marcus. 1995. Text chunking using transformation-based learning. *CoRR*, cmp-lg/9505040.

Pontus Stenetorp, Sampo Pyysalo, Goran Topić, Tomoko Ohta, Sophia Ananiadou, and Jun'ichi Tsujii. 2012. brat: a web-based tool for NLP-assisted text annotation. In *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 102–107, Avignon, France. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *CoRR*, abs/1706.03762.