Entity and Semantic Relation Extraction using CRF and BERT SemEval2021 Task 8 - MeasEval - Counts and Measurements

Sandeep Kothapalli*, Pericherla Amshumaan Varma*, Yeshwanth Moganti*, K V R K Vivek*, and Pedavegi Sasinandan*

* Department of Computer Science and Engineering, Indian Institute of Technology, Kharagpur

Abstract

This paper depicts our system for SemEval2021 Task 8 - MeasEval - Counts and Measurements. [1] MeasEval is a new entity and semantic relation extraction task focused on finding counts and measurements, attributes of these quantities, and additional information including measured entities, properties, and measurement contexts. Our system only deals with two (1 and 3) of the five sub tasks of MeasEval. Our system uses CRF based model for subtask 1 and subtask 3. And we give a BERT based implementation, using Adaptive thresholding, which we believe has potential to perform even better given enough data.

1 Group Details

CodaLabs ID : **iitkgp_cs60075_team9**	
Name	Roll number
Kothapalli Sandeep	17CS10021
K V R K Vivek	17CS10024
M Yeshwanth	17CS10027
Pedavegi Sasinandan	17CS10032
P Amshumaan Varma	17CS30025

2 Individual contributions

2.1 Kothapalli Sandeep (17CS10021)

Subtask Assigned : Subtask 1

Contribution : Did a literature survey for selecting an appropriate model, feature set and their corresponding labels to train the model on, collaboratively with Amshumaan [2.2]. Authored the boilerplate code, including pre-processing of the raw dataset and modifying it appropriately into a format suitable to work while keeping the format required by the competition in mind. Authored the code for extracting the features from the pre-processed data and labels corresponding to each data point from the .tsv files with constant inputs from the team. Collaborated with Amshumaan [2.2] for extending the subtask 1 implementation to subtask 3 for multi label prediction [3.3]. Co-authored the code for training, predicting, validating the model and converting the results into a format suitable for submission. Collaborated with Yeshwanth [2.3] multiple times for bug fixing, model tuning. Co-authored the code for pre-processing the data and extracting features and labels for model 2 [3.4] in sub task 3.

2.2 P Amshumaan Varma (17CS30025)

Contribution : Did a Literature Survey on sequence tagging for appropriate model selection. Authored the code for subtask 1 to extract potential quantites using crf model with constant inputs from Sandeep [2.1]. Authored the code for pre-processing the data and creating feature vectors for crf model input and training the crf model using randomized search for hyper-parameter optimization. Extended this code for subtask-3 multi-label prediction with the help of Sandeep [2.1] and wrote the code for linking the potential entities extracted with their respective quantities extracted in subtask 1 and creating the tsv files for submission. Collaborated with the team for tuning the models and bug fixing.

2.3 M Yeshwanth (17CS10027)

Subtask Assigned : Subtask 3

Contribution : Did a comprehensive literature survey on Sequence tagging [2]. Understood the literature survey done by Sasinandan [2.5] and developed a BERT based model based on [6] for relation extraction. Authored the code for preprocessing the train and test data to be given as inputs to the BERT based model along with inputs and help from Sandeep [2.1]. Debugged the error the

[1] CodaLab - Competition

validation code is giving on the output files format with help from Sandeep [2.1]. Authored the code for the training of the classifier networks and the loss function.

2.4 K.V.R.K.Vivek (17CS10024)

Subtask Assigned : Subtask 3
Contribution : Did a literature survey on Sequence tagging [2]. Understood the literature survey done by Sasinandan [2.5] and using this helped developing a classification problem on relations for the subtask3. Co-authored the code for a training of neural networks, BERT-based model for subtask3. Co-authored the code for predicting the relations between quantities extracted in subtask1 and the potential entities or properties along with Sasinandan [2.5] and helped in debugging the code many a times while implementing the model using pytorch. Collaborated with the team multiple times for bugfixing and model tuning.

2.5 P Sasinandan (17CS10032)

Subtask Assigned : Subtask 3
Contribution : Did a comprehensive literature survey on Relation Extraction[6] and using this proposed a classification problem on relations for the subtask3. Co-authored the code for the training of neural networks, BERT-based model for subtask3. Authored the code for evaluating and extracting the relations between quantities extracted in subtask1 and the potential entities or properties for results.

3 Approach/Model Architectures

3.1 Subtask 1

Conditional Random Field (CRF) [4] is a probabilistic discriminative model (special case of Markov Random Field) that is very useful in Sequence labelling tasks which are widely used for applications in Natural Language Processing, Computer Vision and Bio-informatics. The conditional random field uses contextual information to add information to the model so that it can make accurate predictions.

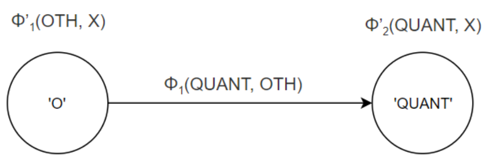


Figure 1: CRF Architecture for SubTask-1

As CRF is a discriminative model i.e it models the conditional probability P(Y/X) where X is always given or observed, the Markov network simplifies to a chain and the training problem reduces to maximizing the log likelihood w.r.t to all model parameters using gradient descent. The model has (M x M)(one for each transition) + (M x S)(for each label) number of parameters) where M is the number of labels to predict and S is the size of the feature vector.

3.2 Subtask 3

For the measured quantities we extracted from subtask 1, we try to extract the measured entity corresponding to each measured quantity and if it exists, we look for the property being measured.

3.3 Model1

We extend our CRF architecture used for subtask 1 to identify the measured entity and property for an extracted quantity in subtask 1. We use an extended range of feature set to include broader windows as unlike quantity measurement, entity and properties have dependency relations with words further away in the sentence. Here there are 4 labels to predict ('OTH', 'QUANT', 'ME' and 'MP').

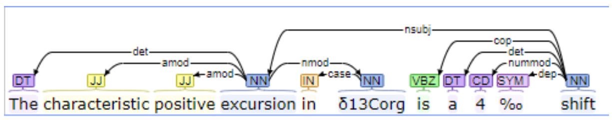


Figure 2: Basic Dependencies as generated by CoreNLP package

As there is no attention involved in the CRF architecture for linking the extracted entities and properties with the measured quantities in subtask 1, we tried to extract the dependency relations between the quantity and entity on the train data using CoreNLP library [3] but there were only very few direct dependencies between them. So we check if the extracted entity or property has any dependency from the extracted list with any quantity in the same sentence. If present, we assign them in the same **annotSet**. Otherwise we link the extracted entities with the closest quantities in the sentence.

3.4 Model2

We wanted to make use of pre-trained embeddings to get a good contextual representation for the potential entities in a sentence. This helps us to get a good knowledge of the dependencies between quantity and entities. We are considering Measured Properties also to be entities having Measured Property relation with a quantity. We follow the assumption that related tuples of the measured quantity, measured entity, and measured property lie in the same sentence.

Upon observing closely, the problem is like relation extraction between an entity and a quantity. We observed how [6] made use of pre-trained models in the task of relation extraction and decided to use BERT to get contextual representations for entities and quantities. One challenge we encounter here is that they can be of multiple tokens. So, to get a single representation, we introduced a special token * before and after an entity or a quantity and took the representation of the * before the phrase as used in [6]. Apart from these contextual representations, we also think that the attention which a quantity exerts on an entity also determines how those two are related. So, for a pair (E, Q), where E is an entity and Q is a quantity, we also took the attention values of Q on E and vice versa in the BERT model as shown in fig.3.

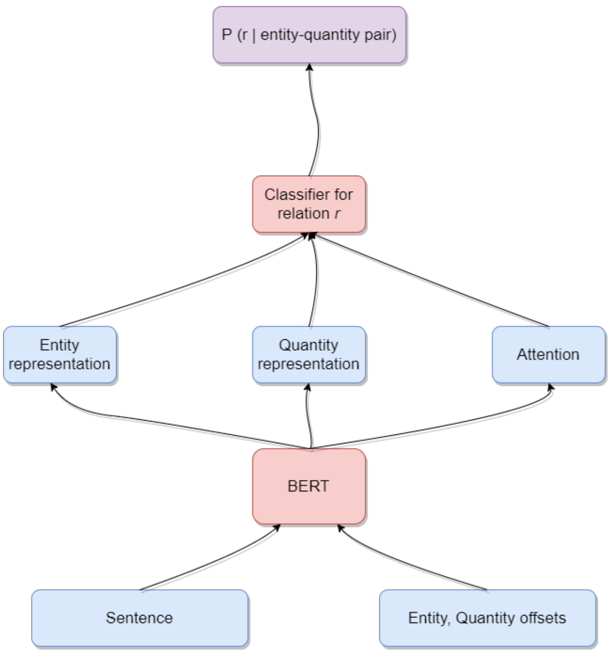


Figure 3: BERT based architecture for subtask 3

We have 3 relations overall - entity relation, property relation, and threshold relation. We are going to use a binary classifier for each relation and it outputs the probability that such a relation exists between the given entity and quantity. The threshold classifier decides an adaptive threshold probability for both the classifiers to predict a yes specifically for each pair instead of a global threshold which can reduce decision errors. During training, each of these classifiers is trained with an adaptive loss function[6] such that the threshold value decides the existence of either of the relations.

To make better predictions, we have to pull the logits of positive relations above the threshold and those of negative relations below the threshold. So, we are using the adaptive thresholding loss as described in [6]. It can be broken down into two parts.

$$L_1 = -\sum_{r \in P_T} log(\frac{exp(logit_r)}{\sum_{r' \in P_T \cup \{TH\}} exp(logit_{r'})})$$

$$L_2 = -log(\frac{exp(logit_{TH})}{\sum_{r' \in N_T \cup \{TH\}} exp(logit_{r'})})$$

$$L = L_1 + L_2$$

Where T denotes the given pair (E, Q) and P_T is the set of relations existing between them and N_T is the set of relations not existing between them, excluding the thresholding relation TH.

While evaluating, we group the pre-processed dataset by quantity and for each quantity, we compare the probability of the existence of entity relation with the corresponding threshold value and classify the existence of a measured entity relation over all the potential entities for a measured quantity. In case of the existence of a measured entity relation, we classify the existence of a measured property relation.

4 Experiments and Results

4.1 Subtask 1

For extracting the quantities, one major feature is whether if its a unit. So we use an available list of units in 'quantities' library to add this as one of the feature. We use other features for each word (generated using spacy [1] etc) like lemma, postag, dependencies, number-like, is digit, nouns and entities it has dependencies with as detected by spacy etc. As its a sequential task, we added a neighbour window of size 2 to add these neighbour properties to the feature vector.

We used crfsuire library [5] provied by sklearn to train implement and train the discussed crf model. For parameter eatimation, we opted 'l-bfgs' algorithm and for selecting the hyperparametes, a randomized search is performed over a scale of parameters for better model selection. It achieved

```
Submission count of Quantity: 458
True positives (matching rows): 445
False positives (submission only): 31
Precision: 0.9348739495798319
Overall Score F1 (Overlap): 0.128419825506403
```

Figure 4: Results for Sub-Task 1 using CRF model. Results shown are obtained by executing evaluation script provided by MeasEval

satisfactory precision on gold data for subtask 1.

4.2 Subtask 3

4.2.1 Model 1

We add few additional features for this task. From the results of subtask-1, we add a new feature 'is-quantity' and also increase the window size to capture wider dependencies as we believe it makes for better model for detecting entities. We implement this similar to the subtask 1 using crfsuite library and performing randomized search for hyper parameter optimization. We can look at the results obtained in Fig 5

```
Precision: 0.6653306613226453
Recall: 0.31841432225063937
F-measure: 0.4307027027027027

Overall Score Exact Match: 0.21824193992835492
Overall Score F1 (Overlap): 0.25703671345988954
```

Figure 5: Results for Sub-Task 3 using CRF model. Results shown are obtained by executing evaluation script provided by MeasEval

We can also look at top transition in the crf model among the labels and few feature weights for predicting each lable as estimated by the model in the following figures.

```
MP     -> MP      2.603903        ME     -> QUANT   -1.775197
QUANT  -> QUANT   2.321604        MP     -> ME      -1.782312
ME     -> ME      2.017225        QUANT  -> O       -1.813423
O      -> O       1.687054        ME     -> O       -2.488604
ME     -> MP      -0.516226       O      -> ME      -2.516852
```

Figure 6: Most (a) Likely and (b) Unlikely Transition in the CRF model

4.2.2 Model 2

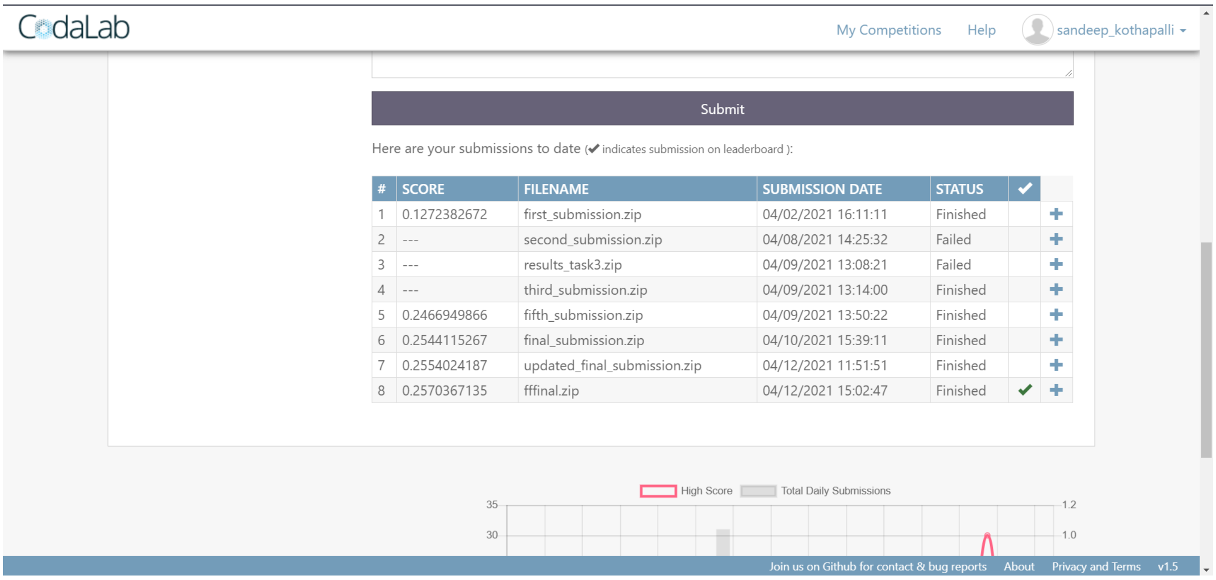
We used pretrained embeddings, we made some experiments by tweaking the learning rate. We used the Stochastic Gradient Descent(SGD) optimizer for the three classifiers. Initially we set the learning rate to 0.01. With that learning rate, the loss function converged very quickly. Then we decreased the learning rate, first to 0.005 and then to 0.0005. It did not converge in either of the cases. Then we looked at the results after prediction, which were not that great. But we think the BERT model with proper tuning and modification, has the potential to perform better. This model has the potential to be a better model but there is no sufficient data to train as we can observe from the hyperparameter variation.

```
5.510045 QUANT   np:≥                2.668392 MP   +1:postag:SYM/NFP
3.220105 QUANT   np:±                2.124177 MP    postag:VBD/FW
2.565138 QUANT   -1:dep:meta         2.053880 MP   -1:postag:WDT
2.471956 QUANT   postag:JJR          1.411470 MP    dep:csubj
2.389043 QUANT   postag:CD
2.356225 QUANT   postag:ADD          1.986182 ME   +1:postag:NFP/agent
2.199326 QUANT   -1:postag:SYM       1.916725 ME    dep:neg
2.076113 QUANT   dep:auxpass         1.760992 ME   -1:postag:WRB/neg/POS
1.871542 QUANT   -1:dep:quantmod     1.339019 ME    postag:NNS
```

Figure 7: Feature weights estimated by the CRF model for different labels

5 CodaLabs submission

GitHub repository : link.
GitHub user **iitkgp-nlp-pg** has been added as a collaborator. Our scores in CodaLab are as shown in figure 8



#	SCORE	FILENAME	SUBMISSION DATE	STATUS	✔
1	0.1272382672	first_submission.zip	04/02/2021 16:11:11	Finished	
2	---	second_submission.zip	04/08/2021 14:25:32	Failed	
3	---	results_task3.zip	04/09/2021 13:08:21	Failed	
4	---	third_submission.zip	04/09/2021 13:14:00	Finished	
5	0.2466949866	fifth_submission.zip	04/09/2021 13:50:22	Finished	
6	0.2544115267	final_submission.zip	04/10/2021 15:39:11	Finished	
7	0.2554024187	updated_final_submission.zip	04/12/2021 11:51:51	Finished	
8	0.2570367135	fffinal.zip	04/12/2021 15:02:47	Finished	✔

Figure 8: Submission #1 is the result after doing only subtask 1 and Submission #8 is the result after doing subtask 3

References

[1] Matthew Honnibal and Ines Montani. "spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing". To appear. 2017.

[2] Onur Kuru, Ozan Arkan Can, and Deniz Yuret. "Charner: Character-level named entity recognition". In: *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*. 2016, pp. 911–921.

[3] Christopher D. Manning et al. "The Stanford CoreNLP Natural Language Processing Toolkit." In: *ACL (System Demonstrations)*. The Association for Computer Linguistics, 2014, pp. 55–60. ISBN: 978-1-941643-00-6. URL: http://dblp.uni-trier.de/db/conf/acl/acl2014-d.html#ManningSBFBM14.

[4] Charles Sutton and Andrew McCallum. "An Introduction to Conditional Random Fields". In: *Found. Trends Mach. Learn.* 4.4 (Apr. 2012), pp. 267–373. ISSN: 1935-8237. DOI: 10.1561/2200000013. URL: https://doi.org/10.1561/2200000013.

[5] Jan Wijffels and Naoaki Okazaki. *crfsuite: Conditional Random Fields for Labelling Sequential Data in Natural Language Processing based on CRFsuite: a fast implementation of Conditional Random Fields (CRFs)*. R package version 0.1. BNOSAC, 2007-2018. URL: https://github.com/bnosac/crfsuite.

[6] Wenxuan Zhou et al. "Document-Level Relation Extraction with Adaptive Thresholding and Localized Context Pooling". In: *arXiv preprint arXiv:2010.11304* (2020).