Project#2: MeasEvals - Counts and Measurements

Joydeep Mondal[1], **Kshitij Jaiswal**[2]
[1]20111266, [2]150340
[1]CSE, [2]MSE
{joydeep20, kjaiswal}@iitk.ac.in

Abstract

Scientific discourses are all about numbers and figures apart from equations and theorems. While it is easier to extract all the measurements and counts from the text, their semantic relations are still not extracted efficiently. This project works towards the problem where text can be ambiguous and inconsistent and the location of this information relative to the measurement or count can vary greatly.
The problem has been divided into two major sections, Entity Recognition and Relation Extraction. The report further talks about how we have used the implementations of Stanza and Conditional Random Fields (CRF).
The data set has been taken from this link uploaded for the competition.
Our main task is to find counts and measurements, attributes of these quantities and additional information including measured entities, properties and measurement contexts.

1 Introduction

The scientific text comprises of a lot of counts and measurements to convey the facts and figures about a topic. Quantification is indeed the best possible way to compare and understand a field of research. Extracting these quantified structures with relevant knowledge about their context becomes essential in understanding and comparing several such studies.

Almost all the prior works tried to solve these tasks as entity extraction and relation establishment task. They proposed different methods for the above mentioned tasks. For entity extraction, Conditional Random Field (CRF) was one of the main component used in majority of them. Some of them have used rule based regular expression, pattern matching, POS tagging and dictionary.

We tried different approaches and compared them to select the best one based on the accuracy metrics. We did feature engineering from the data set we were provided. We used the pre-trained POS tagging based pattern matching algorithm on top of it as well. As we were provided with very less amount of training data we used both, the dev and validation set to train the final model and then tested the model on the test data.

The report has been divided into further sections as problem definition, related research work, corpus description, proposed approach, experiments and results, error analysis, individual contribution, conclusion, references and appendix.

2 Problem Defintion

- The problem has five sub-tasks which include span extraction, classification and relation extraction, including cross-sentence relations. The paragraphs are a part of scientific text and we need to perform the following tasks: (Cod)

    1. Identify all Quantities in the text, specify if they are counts or measurements and identify their spans in the text.
    2. For measurements, identify the unit. For both counts and measurements, classify additional value information (count, range, approximate, mean, etc.).
    3. For both counts and measurements, identify the MeasuredEntity, if one exists. If a MeasuredProperty also exists, mark its span.
    4. Identify the location of ”Qualifiers” to record any additional related context that is needed to either validate or understand the observed count or measurement.
    5. Create relationships between Quantity, MeasureEntity, MeasuredProperty and Qualifier spans using the HasQuantity, HasProperty and Qualifies relation types.

- Major terminologies and definitions
    1. HasQuantity : This relation can be between
        (a) Quantity and MeasuredProperty
        (b) Quantity and MeasuredEntity (if MeasuredProperty is absent)
    2. HasProperty: This relation is between
        (a) MeasuredEntity and MeasuredProperty
    3. Qualifies : This relation is between quantity and Qualifier

- From the problem statement it could be inferred that the task that we need to perform are Entity Recognition along with Classification and Relation Extraction.

3 Related Work

One of the main tasks of this project is measurement extraction along with the corresponding measured entities from scientific articles. We studied and found some of the notable works which may help us to lead our path towards solving it. Rule based regular expression defined by (Sevenster M) for measurement extraction on top of trained Conditional random field (CRF) model to extract measured entity (Bozkurt Selen). Input to this CRF model is Parts-of-speech (POS) tagging and dictionary maps of the words in the sentences from which the measurements have been extracted. The authors claim to achieve high True Positive (97%), Low False Positive (4%) and zero False Negative. We found this approach was not generalized enough, rather very specific to radiology reports. For obvious reasons, it may not work well in general scientific reports. The test set used in this was very small (around 100 files). So, evaluation matrix doesn't hold much confidence either. But the pipeline is well defined and can be used. Another CRF model-based approach to extract the measurements along with their units in (10., 2010). The authors have tried and tested (qua) as well but they found that the recall is very poor. They have also tried OIE (open information extraction)(OIE), Named Entity Recognition (NER) as well but finally went with Grobid's quantities (Lopez, 2010). They also Created Dependency graph for each sentence having the measurement and applied pattern recognition to identify the measured entity along with its property using Stanford Core NLP. This tool is known as

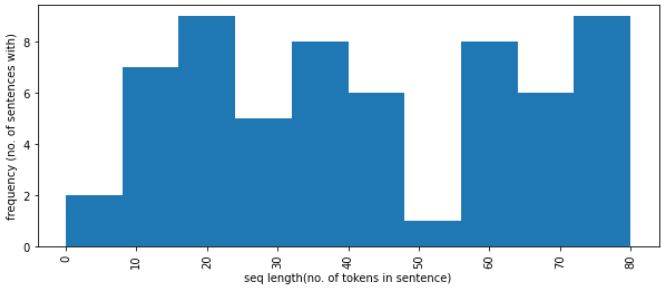


Figure 1: Statistics of Tokens

Marve (Hundman and Mattmann, 2017). The authors claim to achieve Precision of 80.4%, Recall of 66.2%, F-score of 72.6%. We found that this has been done on more generalized settings. Marve extraction came out from Nasa's HyspIRI mission (hys) and is being used by NASA in several application.
Another approach is tried using Bidirectional LSTM-CRF Models for Sequence Tagging by (Huang et al., 2015). They used Bi-LSTM model for sequence input. This helped them generating the contextualized representation for every word. Further classification was carried out using Conditional Random Fields. CRFs have been traditionally used in NER tasks. The F-score achieved using this approach on CoNLL data set is 84.26 (with random embeddings) which is higher than other traditional methods like CRFs,HMMs.

4 Corpus/Data Description

The data we took is a part of the competition at Codalab organised by Elsevier Labs and INDE lab at the University of Amsterdam. Thus we had no control over the volume, type and annotations.

The data set provided by the TA for this course work was divided into two directories, train and dev. We worked on the train part to select our model. train had two directories, namely text and tsv. Here text had all the content in form of paragraphs and tsv had all the classifications and relations. It has 222 text and corresponding tsv files. The dev directory was similar to the train directory and it is used to validate the trained model. It has 28 text and corresponding tsv files. The tokens 1 in each file do not exceed 80 by any means.

We also observe that the Qualifier Tag is very small in number, frequency of any of the given tags doesn't exceed 1000 and the entire data set is very small.

We were later provided with a test directory

Table 1: Frequency of Tags

Data Statistics		
Tag	**Freq. Train**	**Freq. Dev**
MeasuredEntity	789	71
MeasuredProperty	380	46
Qualifier	204	12
Quantity	741	89

which had 28 files each of tsv and text. This is the data-set on which our model got evaluated.

Observation from Data: We observed a lot of inconsistencies with the data. At some places '%' was not tagged for percentage unit, "20 x 20 degrees" was tagged as list of degrees, 'one', "30 degrees", large integral values which weren't dates/years and other similar data were left untagged and large numbers appeared with and without ',' while no white space was found around several quantities. A reason for lower frequency of Qualifiers can be attributed to the fact that very few of these were recognised in the data set despite their presence.

5 Proposed Approach

In order to solve the problem we followed the following pipeline.

1. **Preprocessing:** We converted the text from the documents into sentences and then into tokens. Thhese tokens were given 'pos' (part of speech) tags from NLTK and corresponding 'ner' (Named Entity Recognition) tags from Stanza.

2. **Measurement Extraction:** Quantity Identification is the first task in which we needed to extract the measurement or quantities. This has been done using several features like 'ner' (gives ner tag), 'word.lower()' (checks whether the all the characters in the word are in lower case), 'word[-3:]' (last three characters in the word), 'word[-2:]' (last two letters of the word), 'word.isupper()' (checks if the word is upper cased), 'word.istitle()' (Checks if the first letter is upper cased), 'word.isdigit()' (Checks if the word is digit), 'postag' (pos tag of the word), 'postag[:2]' (first two characters of the tag) of the word in consideration and the words before and after it.

3. **Measurement Classification** : After quantity extraction there is a subtask in which the extracted quantities need to be classified accordingly. We are treating categorization of the extracted quantities as a Multi-label Classification problem where we classify Quantities in 11 different classes, namely: 'HasTolerance', 'IsApproximate', 'IsCount', 'IsList', 'IsMean', 'IsMeanHasTolerance', 'IsMeanIsRange', 'IsMedian', 'IsRange', 'IsRangeHasTolerance' and 'Unknown'.
Feature Extraction: We have extracted eighteen features 2. After feature extraction we applied traditional machine learning models for classification and Naive Bayes classifier provided the best result for us on the dev set.

4. **Measured Entity, Property & Qualifier Extraction :** After the quantity extraction and classification, measured entity, property and qualifier needed to be extracted. The features used for these tasks include nearest previous Quantity, nearest next Quantity, possibility of the word being a Quantity, distance from previous Quantity, distance from next Quantity, word tag, possibility of the word being an Entity along with all the features used in Quantity extraction sub-task.

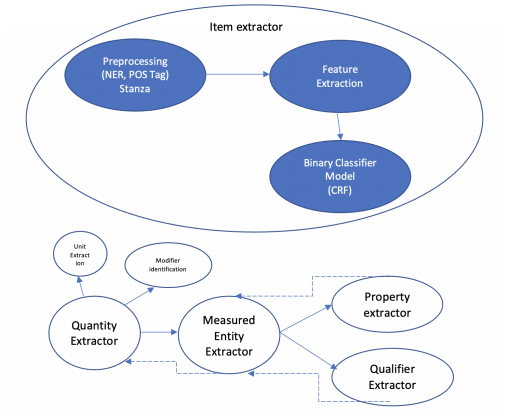
5. **Relation Establishment and Refinement:** Post entity extraction, relation among those extracted quantity, entity need to be established. We went for a deterministic approach here. As we sequentially extracted quantity and associated entity for the already extracted quantity, followed by associated property and qualifier corresponding to the already extracted entity. Now, Once we extracted these associated items we established "HasProperty" relation between the extracted property, extracted entity and "HasQuantity" relation between extracted property and quantity and "Qualifies" to extracted qualifier and extracted entity. In case property couldn't be extracted we established "HasQuantity" relation between extracted quantity and extracted entity.

6 Experiments and Results

Key points (Following can be in any order):

1. As we mentioned in our related work, we tried quantity extraction with Grobid's Quantity as

Figure 2: Overall Architecture



our baseline which provide a F-measure of 0.56 for our training data. We used pretrained model for Grobid. Our best model F-measure on dev data set came out to be as 0.874. For all our experiments, we used python notebooks (Python3) as our programming language and standard python libraries. Preprocessing took the majority of the execution time in the whole pipeline as it required to extract NER and POS tag using pretrained Stanza and NLTK models. Our code is available at https://github.com/joy-deep-cs/CS779CourseProject.

2. Dev sets Results are reported in 3 and test sets results are reported in 4

3. Results of the competing group is reported 5. They have only provided evaluation results for 6 subtasks (quantity, entity, property, hasquantity, hasproperty, unit) out of 9 (quantity, entity, property, qualifiers, hasquantity, hasproperty, qualifies, unit, modifiers).

7 Analysis

1. We noticed that this tasks are highly dependent on the volume of the training datasets. We tried to train the Bert model with the training data we had and it failed miserably. We got close to zero F measures. Then we used pretrained models of Stanza which produced best result for us. Also with the very small number of test data (28 test files), it is very difficult to compare two models.

2. CRF model suffers from low availability of data provided. Model has high training F-measure shown in 3 for Quantity but low test F-measure as shown in 4. This shows very high over-fitting of models. With more training data available, this error can be taken care of. The F1 overlap is considerably low as well.

8 Individual Contribution

Individual Work	
Name	**Tasks**
Joydeep	Literature Review for quantity extraction, Test data accuracy metric with grobid's quantity, Git codebase skeleton setup, CRF model training and prediction for all the sub tasks, model evaluation
Kshitij	Literature Review for quantity classification, Quantity classifiaction, Stanza implementation for preprocessing, project documentation and report management

9 Conclusion

The project work for this course has been really helpful in learning a lot of key concepts. As we can see that the F1 scores are a bit poor for the various tasks. Having spent considerable amount of time we feel our current model needs a customised pos and ner tagging for the tasks. That is something we look forward to. Also Unit can be mapped properly by using a dedicated lexicon that maps unit with their properties. Qualifiers and Qualifies were also showing poor results but this can also be attributed to their limitation in the training data set. Deep Network models perform best when the data is in large amounts. We did not opt for data augmentation but tried transformer based models which gave very poor F1-scores, though their accuracy was good.

References

SemEval2021 Task 8 - MeasEval - Counts and Measurements. [link].

https://www.quantalyze.com/. [link].

https://nlp.stanford.edu/software/openie.html. [link].

https://hyspiri.jpl.nasa.gov/. [link].

2010. *IRFC'10: Proceedings of the First International Information Retrieval Facility Conference on Adbances in Multidisciplinary Retrieval*. Springer-Verlag, Berlin, Heidelberg.

Banerjee Imon Rubin Daniel L. Bozkurt Selen, Alkim Emel. Automated detection of measurements

and their descriptors in radiology reports using a hybrid natural language processing algorithm. *Journal of Digital Imaging*.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional lstm-crf models for sequence tagging.

Kyle Hundman and Chris Mattmann. 2017. Measurement context extraction from text: Discovering opportunities and gaps in earth science.

Patrice Lopez. 2010. Automatic extraction and resolution of bibliographical references in patent documents. In *Proceedings of the First International Information Retrieval Facility Conference on Adbances in Multidisciplinary Retrieval*, IRFC'10, page 120–135, Berlin, Heidelberg. Springer-Verlag.

Liu P Peters JF Chang PJ Sevenster M, Buurman J. Natural language processing techniques for extracting and categorizing finding measurements in narrative radiology reports. *ACI - Applied Clinical Informatics*.

Table 2: Justification for Feature Extraction

Justification for Feature Extraction		
Feature	**Reason**	**Method**
Unit Features	To differentiate between Date, Measurements and Counts	From "other" column we extracted the value for key "unit" ("Unknown" where absent) and mapped with corresponding numeric value.
Total Digit Count	To find total number of digits in text	Made a list of all numbers in text and counted relevant digits
Number Count	To find presence of list in text	We counted the length of list made in previous exercise
Average Digit Count	To find average length of each numeric value given	We divided Total Digit Count by Number Count
Delimiter Present	To find presence of list	We searched in a dictionary for the presence of 'and', '&', 'or',',' and ';'.
Float Flag	To find presence of float type of values	Flag was turned on when decimal value was found
Length	To find length of text considered	We found the difference of 'endOffset' and 'startOffset'
Category-wise Word Feature (11 features)	To map word wise categories	We found the exhaustive set of valid English Word for each category and then mapped its presence from the text

Table 3: Results on dev-set

Result on dev set				
	Precision	**Recall**	**F-measure**	**F1 Overlap**
Quantity	0.870	0.879	0.874	0.692
Measured Entity	0.149	0.134	0.141	0.064
Measured Property	0.050	0.017	0.025	0.013
Qualifier	0	0	0	0
Unit	0.455	0.407	0.429	0.273
Modifier	0.053	0.135	0.076	0.039
Has Quantity	0.078	0.072	0.074	0.039
Has Property	0	0	0	0
Qualifies	0	0	0	0

Table 4: Results with our model on test-set

Results with our model on test-set				
	Precision	**Recall**	**F-measure**	**F1 Overlap**
Quantity	0.907	0.719	0.803	0.616
Measured Entity	0.139	0.136	0.137	0.063
Measured Property	0.133	0.036	0.056	0.025
Qualifier	0.062	0.04	0.049	0.025
Unit	0.36	0.3	0.327	0.195
Modifier	0.064	0.133	0.087	0.045
Has Quantity	0.085	0.085	0.085	0.044
Has Property	0.083	0.018	0.029	0.015
Qualifies	0	0	0	0

Table 5: Results of Competitor's on test-set

Results of Competitor's on test-set				
	Precision	**Recall**	**F-measure**	**F1 Overlap**
Quantity	0.793	0.890	0.839	0.700
Measured Entity	0.313	0.423	0.36	0.179
Measured Property	0.125	0.053	0.074	0.023
Qualifier	0	0	0	0
Unit	0.836	0.767	0.8	0.667
Modifier	0	0	0	0
Has Quantity	0.078	0.108	0.091	0.047
Has Property	0.042	0.018	0.025	0.013
Qualifies	0	0	0	0

Table 6: Overall Performance Comparison

Overall Mode Comparison		
(Gold count)	**Our**	**Others**
Quantity (82)	62	75
Measured Entity (81)	79	115
Measured Property (51)	15	24
Qualifier (25)	16	0
Precision	0.259	0.408
Recall	0.208	0.333
F-measure	0.231	0.367
F1 overlap	0.122	0.211