Extracting Numeric Assertions from Text

Anonymous ACL submission

Abstract

Open-domain Information Extraction (IE) plays an essential role in constructing large-scale knowledge bases and supports downstream applications such as Question Answering, Text Summarization, etc. While most prior research in IE has centered around extracting categorical relational tuples (e.g., president of, located in), the extraction of numerical relations (e.g., literacy rate, area, molecular weight), that link quantitative mentions to corresponding entities, remains relatively underexplored. This work addresses this gap by targeting the extraction of open-domain numeric assertions, which require identifying both the relevant entity and the appropriate measuring attribute associated with a quantity in natural language text. We begin by refining an existing OpenIE system through a rule-based approach where retrieving implicit measuring attributes for a quantity mention becomes the main challenge. To overcome this, we propose a neural framework that jointly identifies the relevant entity for a numeric mention and infers the measuring attribute to relate them, using contextual cues in the sentence. Experimental evaluation shows that our proposed model outperforms the baseline and a general-purpose large language model with a significantly large margin.

1 Introduction

Quantitative information is a fundamental aspect of factual knowledge and plays a crucial role in enabling downstream NLP tasks (Mausam, 2016), such as question answering, summarization, recommendation, and fact verification. Numeric assertions, which link entities and quantities with appropriate measuring attributes, provide valuable insights into numerical properties, ranging from population sizes and distances to prices and percentages. Such assertions also enrich knowledge bases and support reasoning over quantitative facts.

Despite the prevalence of numerical content in text, existing OpenIE frameworks (Saha et al., 2017; Cui et al., 2018; Pei et al., 2023) often fail to capture such numeric assertions, specifically extracting the measuring attributes of an entity as a relational phrase. For example, in the sentence "With an estimated spread of 1,774.53 square kilometers, an estimated 111.8 people live per square kilometer in Sinanana Dinsho", existing systems fail to infer that the first quantity refers to the area of Sinanana Dinsho, while the second refers to its population density. While one can consider a simple rule to use units to identify the measuring attribute, no such dictionary is present that covers an exhaustive list of units. Moreover, quantity may be dimensionless, such as the capacity of a playground or the number of habitat. Large Language Models (LLMs), like BERT (Devlin et al., 2019), GPT4 (OpenAI, 2023), Gemini (Anil et al., 2023), etc. are very effective in capturing contextual cues, which allows them to generate semantically related measuring attributes from sentences, but they still struggle when cues are implicit or ambiguous. For instance, given the sentence "Hyundai owns 33.88% of Kia", GPT-4 often fails to generate the precise measuring attribute share, instead producing semantically broader terms like ownership. However, very few systems extract numerical information, but model them for the information retrieval task (Ho et al., 2019; Alonso and Sellam, 2018), and therefore focus on retrieving the context rather than specific measuring attributes.

This work focuses on extracting numerical assertions from sentences, specifically extracting a measuring attribute as a relational tuple, where the object is a quantity. Here, we aim to extract triples ⟨E, M, Q⟩ from a sentence, where E is an subject entity, M is the associated measuring attribute, and Q is the corresponding numerical value with the unit if available. For example, from a sentence "Arlescote is fairly remote, with the nearest village being 2 miles away", the valid extraction is ⟨**E**: 'Arlescote', **M**: 'distance from nearest village',

Q: '2 miles'⟩. To extract the numerical assertions from sentences, we propose a neural framework that jointly identify the entity and the measuring attribute for a given quantity in a sentence. As there is no annotated dataset for numerical triples available in literature for training the proposed network, we first generate a large-scale pseudo-labeled dataset from Wikipedia using a rule-based approach that converts triples extracted from an existing OpenIE system to acquire labeled triples. We summarize our contributions as follows.

- We develop a rule-based approach to generate numerical triples from Wikipedia on top of an existing OpenIE system.
- We introduce a neural framework to extract numerical assertions, trained on the dataset generated by the proposed rule-based pipeline.
- A comparative evaluation of the proposed models against a few baselines is reported.
- The codes and data generated from the rule-based model are made public here and the neural models are available here.

2 Related Works

Open Information Extraction. Early OpenIE systems, such as KnowItAll (Etzioni et al., 2004), TEXTRUNNER (Yates et al., 2007), REVERB (Etzioni et al., 2011), use human-defined patterns and sentence structure to generate facts from web data. OLLIE (Mausam et al., 2012) overcomes the drawbacks of REVERB and takes advantage of the pattern learning approach using bootstrapping. While REVERB is a verb-mediated extractor, RELNOUN 2.0 (Pal and Mausam, 2016) is a noun-mediated extractor for complex noun phrases and uses linguistic processing for extraction. Open IE 4.2 [1] combines SRLIE (Christensen et al., 2011) and RELNOUN (Pal and Mausam, 2016).

Recently, encoder-decoder architecture is explored to model the open IE task (Cui et al., 2018). OpenIE6 (Kolluru et al., 2020a) exploits the BERT-based encoder to design an iterative grid-level architecture that imposes constraint rules and conjunction rules to improve the earlier system. DetIE (Vasilkovsky et al., 2022) trains a model using BERT (Devlin et al., 2019) and transformer encoder architecture with order-agnostic loss to improve efficiency of the OpenIE task. A recent survey (Pei et al., 2023) categorizes OpenIE systems and datasets by application. However, all these systems are generic and do not target numeric relations or implicit measuring attributes.

Relation Extraction Models. Relation extraction (RE) is typically framed as a classification task over predefined relations. State-of-the-art RE models leverage pre-trained language models like BERT (Devlin et al., 2019) to capture entity and context representations (Wu and He, 2019; Tian et al., 2021; Li et al., 2019). Aman et al. (Madaan et al., 2016) propose a rule-based relation scoping and probabilistic graphical model to extract numerical relations, like inflation rate, atomic number, etc. While RE models perform well on curated datasets such as TACRED(Zhang et al., 2017) and KB-37 (Zhang and Wang, 2015), they are limited to closed schemas and do not generalize to open-domain numeric assertions. A recent survey (Zhao et al., 2024) presents a range of relation extraction methods, however, none of them specifically target numerical relations.

Extraction of Numerical Assertions. BONIE (Saha et al., 2017) introduces the first numerical assertion extractor, built using the Bootstrapping method. It extracts triples in the form ⟨Arg1, relation phrase, Arg2⟩ where Arg2 captures quantity phrase. While it improves precision over OpenIE 4.2, it struggles with conjunctions and implicit measuring attributes. Many works formulate the information extraction task as a sequence tagging problem (Ho et al., 2019; Kolluru et al., 2020b) where input tokens are tagged according to the required output forms. There are NLP pipelines (Roy et al., 2015) that specifically aim to identify quantities in text; however, they do not retrieve the associated entity or context that describes the quantities. Addressing this problem, a few works explored the task as retrieving related information for a quantity mention in text (Alonso and Sellam, 2018; Ho et al., 2019). However, as the target application is considered open-domain question-answering, they accept noisy or partial extraction. A recent work (Zhang, 2022) targets the extraction of numerical assertions from Chinese text by fine-tuning BERT on question-answering dataset, where they inquire about the entity and the attribute for a target quantity in a sentence.

[1] https://github.com/allenai/openie-standalone

3 Rule-Based Approach for Training Data Generation

We introduce an extraction pipeline to construct a large-scale dataset of numerical triples, in the form of ⟨ E, M, Q ⟩ from natural language text, which we use to train neural models. Our extraction pipeline consists of two steps–1) filtering OpenIE outputs and 2) converting filtered triples to ⟨ E, M, Q ⟩.

3.1 Filtering OpenIE Generated Triples

OpenIE systems are effective in extracting general purpose relational tuples, typically in the form ⟨subject(S), predicate(P), object(O)⟩, but they are not optimized to identify numerical information or associate it with appropriate measuring attributes. Here, we adopt one of the existing OpenIE systems BONIE (Saha et al., 2017) as it aims at retrieving triples containing numerical mention in any of the three arguments in ⟨S, P, O⟩ extracted from a sentence. Hence, we filter the triples extracted by BONIE using the following rules to find potential candidate ⟨S, P, O⟩ triples that are later transformed into numerical assertions.

- We discard triples that do not include any quantity in the object phrase and have a confidence score less than 0.9.
- We eliminate all ⟨S, P, O⟩ triples that specify any temporal tags (e.g., date, seconds, etc.) or geolocation in the subject or object phrase.
- We filter ⟨S, P, O⟩ with multiple quantities in the object phrase as such triples can introduce noisy ⟨E, M, Q⟩ during conversion.

3.2 Rule-based method for converting OpenIE Triples to Numerical Assertions

The retrieved triples may capture the measuring attribute within the subject or predicate phrase. Additionally, measuring attributes (e.g., distance, duration, area) are often implicit or embedded in nominal phrases, and thus are missed out by the extractor. Therefore, we devised a rule-based approach to convert the extracted triples ⟨S, P, O⟩ into numerical assertions ⟨E, M, Q⟩ by identifying the quantity Q and measuring the attributes M from the subject, predicate, or object phrase of extracted triples by BONIE.

To facilitate effective conversion, we first determine whether the sentence associated with a given ⟨S, P, O⟩ triple matches one of the following patterns, defined below.

Sentence Pattern 1: It captures the sentences for which the predicate (P) extracted by BONIE contains the verb that is also the syntactic root. Additionally, the subject phrase (S) must contain at least one named entity, and the noun phrase in S or P (identified by a PROPN or NOUN) is linked via 'nsubj' or 'dobj' to root, respectively.

Sentence Pattern 2: It considers all sentences that do not follow the syntactic structure of pattern 1, but the extracted S, P, and O contain a noun phrase, only a verb, and a quantity without any noun phrase, respectively.

For both sentence patterns, we extract the quantity (Q) from the object phrase (O) using the QUANTULUM3 parser (Mündler, 2022). It identifies the numerical expressions and their associated units from the extracted object phrase in ⟨S, P, O⟩ by BONIE. Finally, we apply a set of pattern-specific rules to retrieve the entity and the measurement attribute.

3.2.1 Extraction Rules for Sentence Pattern 1

We apply the following set of rules to extract the subject entity (E) and the measuring attribute (M) from the ⟨S, P, O⟩ and the source sentence that matches pattern 1.

Extraction of Entity(E): We identify the subject entity using the syntactic structure of the source sentences and the entities named in the subject phrase (S) in ⟨S, P, O⟩ triples. For this, we exploit the following three heuristic cases.

- **case 1:** If a named entity is detected in the subject phrase S using SpaCy NER library, we retrieve all connected noun phrases linked to the entity from the dependency parse tree of the source sentence, and consider them together as entity phrase (E). For example, we extract 'Montreal and the southern Quebec region' as entity phrase E from "Montreal and the southern Quebec region receive slightly over 2,000 hours of sunshine annually, with summer being the sunniest", whereas the BONIE extracts only 'Montreal' as the subject S.
- **case 2:** If the subject contains a single named entity that is a proper noun and there are no linked noun phrases, it is extracted as E.
- **case 3:** If there are no named entities detected by SpaCy in subject phrase S, we identify the

proper noun from it as the entity E. If multiple such noun phrases exist, we choose the last one as the entity E.

Extraction of Measuring Attribute (M): Considering the position of the noun phrase in predicates and objects in ⟨S, P, O⟩ triples, we devise the following rules to identify measuring attributes M.

- **Case 1:** If the predicate has one noun phrase, it is extracted as the measuring attribute M; otherwise, the noun phrase nearest to quantity Q is chosen.
- **Case 2:** When predicate P does not contain any noun phrases, we find all noun phrases in the object O. If there is only one, we consider it as the measuring attribute M. Otherwise, we consider the one closest to the quantity Q as the measuring attribute. For example, we extracts 'surface elevation' as the measuring attribute from triple ⟨ Lake Baikal, has, surface elevation of 455.5 meters⟩ extracted by BONIE from the sentence "Lake Baikal has surface elevation of 455.5 meters".

In both cases, if the noun phrase includes the immediate preposition 'of', we consider the linked phrase with 'of' as part of the measuring attribute. For example, the extracted predicate 'poverty ratio' from "the country had a poverty ratio of 46.86% of the households in 2005." is modified to 'poverty ratio of the households'.

3.2.2 Extraction Rules for Sentence Pattern 2

Sentences that match with pattern 2, we first apply the same entity extraction rule as in Sentence Pattern 1 (Case 1) and extract linked noun phrases with the entity. Next, we use the following rules to retrieve the entity phrase E and the attribute M jointly from the extracted phrase.

- **Case 1:** If multiple noun phrases exist and one of them is possessive, it's taken as the entity E. From the rest, the first noun becomes the measuring attribute M. For example, in "The average age of the Politburo's members was 58 years", the phrase 'Politburo's members' becomes E, and 'average age' is considered as measuring attribute M. Otherwise, the first noun phrase recognized by NER is considered as the entity E. The next noun phrase in the order is selected as the attribute M.
- **Case 2:** If there is a single noun phrase containing a named entity, the detected entity is considered E and rest of the phrase becomes attribute M. For example, in "Chicago's overall sales tax is 9.75% ", we identify Chicago and overall sale tax as the entity E and the measuring attribute M, respectively. Otherwise, that noun phrase is considered as E and the measuring attribute is retrieved by converting the adjective present within a window of two words from the quantity Q to the nominal phrase. If no such adjective exists, the attribute is considered as a null string.

Using these rule-based extractor, we generate a corpus of over 90,000 numerical assertions from Wikipedia sentences, which is employed to train our proposed LLM-based neural extractors, proposed in the next section.

4 LLM-based Extraction of Numerical Assertions

Identifying patterns and creating corresponding rules for extracting ⟨E,M,Q⟩ triples for different types of sentences are labor-intensive and inefficient for diverse unseen sentence patterns. Furthermore, measuring attributes are often indirectly referenced in the sentence. Although, BONIE uses a unit-based dictionary to infer implicit measuring attributes, its effectiveness is limited based on the unit coverage of the dictionary. Even when units are present, they can ambiguously refer to multiple properties (e.g., monetary units may correspond to profit, loss, asset, or net worth). In case of dimensionless quantity, dictionary-based measuring attribute inference is not even applicable. These challenges highlight the importance of contextual reasoning in identifying the measuring attribute. To overcome these limitations, we propose am end-to-end transformer-based neural framework to find numerical assertions from sentences.

We design a multi-tasking learning framework by combining a sequence tagging module built on top of a BERT-based encoder and a generative transformer decoder. The motivation stems from the observation that while entities E are usually mentioned explicitly in sentences and can be effectively extracted via token classification, measuring attributes M often require generation based on implicit contextual cues from sentences.

Quantity-pivoted encoder module. We design this module to identify the entity phrase and to

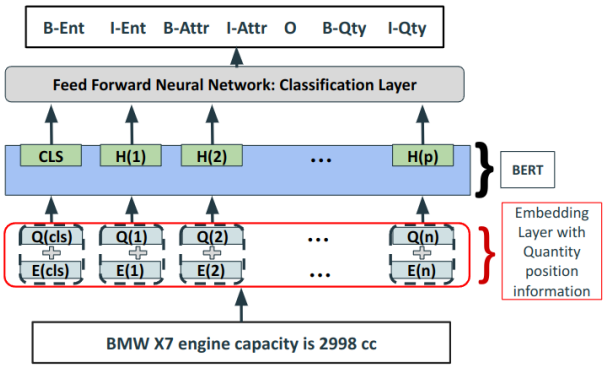


Figure 1: Quantity-pivoted Sequence Tagging Module

capture the cues for generating measuring attribute for a given quantity. Similar to sequence tagging task, the input sentence is fed to BERT encoder along with the position of a quantity mention as a pivot, and each token from the input is classified into three types of tags – Entity, Attribute, and Others, depicted in Figure 1. We use BIO encoding to represent the tags. The quantity embedding layer provides extra information about the position of pivot quantity by using a vector of 1's. Other tokens are encoded with vectors of 0's. Then the average of quantity and token embeddings is passed to BERT encoder. The model considers cross-entropy as the loss function ($\mathcal{L}_{enc}$) to measure the disparity between the actual tags (annotations) and the predicted tags generated by the encoder.

Generative Module. While the entity phrase is always explicitly present in the input sentence, measuring attribute can be implicit or inferred from context, as discussed in earlier sections. Hence, we introduce a decoder for generating measuring attributes considering the contextual cues provided by the encoder. Finally, the output from decoder and encoder can be combined to extract the final assertion. For example, consider the sentence 'Hyundai owns 33.88% of KIA'. The encoder predicts the tag sequence [B-ent, O, B-qty, I-qty, O, I-attr] for the input tokens, identifying 'Hyundai' as the entity and 'KIA' as the attribute. And the decoder generates 'shares of' based on the contextual cues from encoder, resulting in the final assertions ⟨Hyundai, shares of KIA, 33.88%⟩. For training the decoder, we consider the cross-entropy loss between the generated tokens and the target tokens as the decoder loss $\mathcal{L}_{dec}$.

As encoder and decoder modules are performing two different but related tasks, we propose a joint training strategy, allowing the encoder and decoder to be optimized simultaneously. This way the model leverage the information contained in the gradients from the encoder to guide the weight updates on the decoder and vice versa.

We define a unified loss function $\mathcal{L}$, given in Equation 1, as a weighted linear combination of the encoder loss ($\mathcal{L}_{enc}$) and the decoder loss ($\mathcal{L}_{dec}$) mentioned earlier.

$$\mathcal{L} = (1 - \alpha) * \mathcal{L}_{enc} + (\alpha) * \mathcal{L}_{dec} \quad (1)$$

Here, we propose two joint learning frameworks that consider different information flows to share the information between the encoder and decoder.

4.1 Tag-aware Joint Extraction

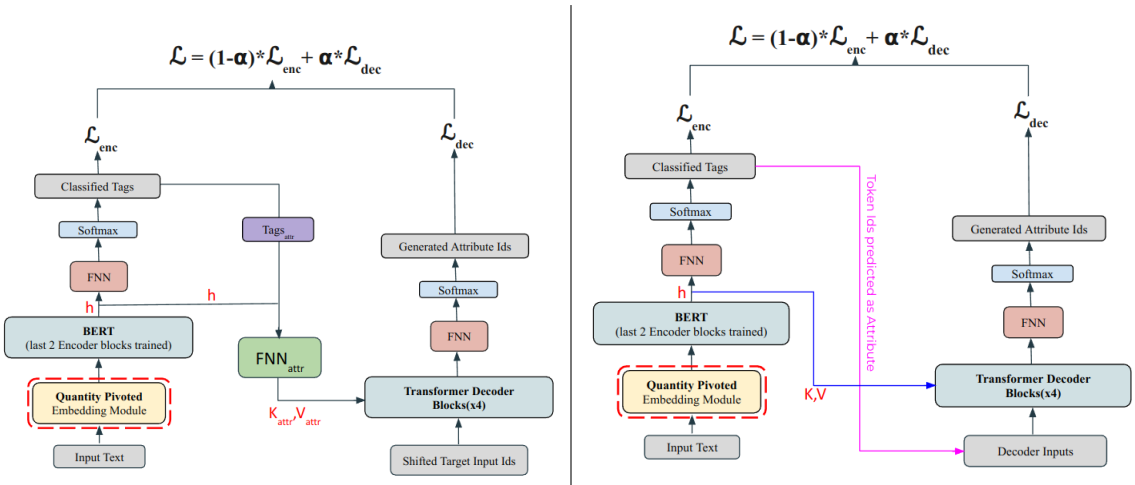
In this approach, we enrich the information flow from encoder to decoder by providing token positions predicted as attributes by the encoder with the cross-attention mechanism. The process involves concatenating the last hidden state embeddings of the encoder with a vector representing the token positions predicted as attributes. This concatenated tensor is then passed through a Linear layer with a rectified linear unit (ReLU) activation function, generating a final embedding of a token. The modified hidden representation is defined as follows:

$$K_{attr} = ReLU(W_{attr}.[h, Pos_{attr}]) \quad (2)$$

Here, K_{attr} represents the Key and Value passed to the Decoder's cross-attention mechanism, h denotes the last hidden state embeddings of the Encoder, and Pos_{attr} is a binary vector, where ones indicate the positions predicted as attribute tag in the sequence tagging module. The framework is illustrated in Figure 2 (a). This enhancement allows the decoder to more effectively attend to semantically relevant regions of the input when inferring measuring attributes.

4.2 Attribute-aware Joint Extraction:

In this variant depicted in Figure 2 (b), we make the decoder explicitly aware of the measuring attribute tokens predicted by the encoder. Unlike the conventional Transformer decoder that typically receives shifted target token IDs as input, we fed the token IDs corresponding to the predicted attribute spans as the query into the decoder. This modification helps the decoder to capture better contextual cues and anticipate relevant measuring attribute. Additionally, the encoder's final hidden state representations are passed to the decoder via cross-attention



(a) Tag-aware Joint method

(b) Attribute-aware Joint method

Figure 2: LLM-based extraction models for numerical assertions

as in the standard Transformer architecture. This setup allows the decoder to condition its generation on both: (i) the structural cues derived from attribute predictions, and (ii) the full contextual information encoded in the source sentence.

5 Experimental Results

Datasets. We constructed a total of 93,769 numeric assertions using our rule-based extractor from Wikipedia sentences which is made public. The dataset statistics are given in Table 1. Since existing OpenIE benchmark dataset do not focus on numerical assertions where the measuring attribute serves as the predicate, we use our dataset generated using the rule-based method as training data. To evaluate the quality of this dataset, we randomly sampled 100 generated triples and assessed how precise the extraction was with precision, reported in Table 2. The results indicate that the rule-based extractor performs more reliably for shorter sentences. Based on this observation, we curated a cleaner subset of 10,000 high-quality triples extracted from sentences of length fewer than 15, which we use to train our LLM-based extraction models. For evaluation of all baselines and the proposed models, we created a test dataset by manually annotated 100 Wikipedia sentences with gold ⟨E, M, Q⟩ triples.

Evaluation Metrics. We primarily use precision, recall and F1 measure to evaluate the quality of the extracted entity and measuring attribute for triples. As these metrics do not consider the word order, we consider BLUE score as well for the evaluation.

Table 1: Statistics of rule-based dataset

Length of sentences	Complete	Test
< 11	5745	25
$11 \leq$sen. len < 21	26985	25
$21 \leq$ sen len < 31	36830	25
≥ 31	24209	25

Table 2: Evaluation of rule-based model using Precision

Sentence Length	E	M	Q
≤ 15	0.96	0.76	0.94
> 15	0.86	0.62	0.82

Experimental Setup The hyperparameters for each model were determined through a 5-fold cross-validation. We set the best value for α that tunes the combined loss to 0.5. The batch size and the learning rate are set to 8 and $1 \times e^{-5}$ respectively. The dropout value for the FNN in the encoder block is set to 0.1 and the dropout for the FNN in the Tag-aware Joint method is set to 0.3. We use 4 decoder blocks from transformer library.

Baselines We employ BONIE as our underline extractor for rule-based model. Therefore, we consider our rule-based approach as the primary baseline, instead of BONIE itself. Furthermore, we use the encoder-decoder architecture (Cui et al., 2018) along with GPT-4 as competitors. Additionally, a fine-tuned GPT 3.5 turbo is also considered as another baseline. We observed that GPT fine-tuning process is very sensitive towards noise and also costly. Therefore, we finetune GPT-3.5 turbo with manually annotated 200 samples, instead of using the automatically generated training data.

Table 3: Performance of models on extraction of measuring attributes

Performance (Sentence Length $\leq$ 15)				
Models	**Precision**	**Recall**	**F1 Score**	**BLUE Score**
GPT 4*	0.71	0.68	0.69	0.41
fine-tuned GPT-3.5	0.85	0.77	0.81	0.64
Encoder-Decoder Arch.	0.58	0.55	0.56	0.53
Rule Based Model	0.71	0.68	0.69	0.58
Tag-aware Joint Extraction	0.91	**0.94**	**0.92**	0.73
Attribute-aware Joint Extraction	**0.92**	0.90	0.91	**0.78**
Performance (Sentence Length > 15)				
GPT 4*	0.60	0.60	0.60	0.27
fine-tuned GPT-3.5	**0.72**	0.63	0.67	**0.40**
Encoder-Decoder Architecture	0.52	0.50	0.51	0.22
Rule Based Model	0.67	0.59	0.63	0.30
Tag-aware Joint Extraction	0.66	0.61	0.63	0.32
Attribute-aware Joint Extraction	0.72	**0.68**	**0.70**	0.31

5.1 Evaluation of he Models on Extracting Measuring Attributes

Table 3 provides a comprehensive evaluation of the extraction methods for measuring attributes. From the results presented in Table 3, we can see that the proposed joint extraction methods significantly outperform all baselines w.r.t. all metrics. We can also observe that tag-aware and attribute-aware extraction methods perform better in extracting measuring attributes from the smaller sentences (length $\leq$ 15) compared to the longer ones. This is consistent with the characteristic of the training data.

Comparing the performance of the proposed models with the encoder-decoder architecture, we can ensure that the availability of additional task-specific information to the decoder from the encoder stack significantly contributed to train a better model for extracting measuring attributes.

We can also observe from the results that the rule-base models perform more steadily across different sentence lengths compare to the proposed neural models. This characteristic reflects that we are able to cover sentences with different lengths with the two patters mentioned in Section 3. However, the overall performance of the rule-based model in extracting measuring attributes is inferior. It is affected by the limitation of BONIE in extracting measuring attributes, as the rule-based model mainly rearranges the extracted BONIE triples $\langle$ S,P,O $\rangle$ to numerical assertions $\langle$ E,M,Q $\rangle$.

5.2 Evaluation of the Models on Extracting the Entity Phrases

Table 2 has shown that the precision in retrieving the entity phrase using the proposed rule-based model is efficient, reaching 96% precision for shorter sentences, specifically due to the precise extraction of entity phrase by BONIE. Overall, we achieve 91% precision for entity extraction using the rule-based model. It is important to note here that the rule-based model is able to detect the conjugated entities where BONIE fails.

From the evaluation, we can also see that the performance of the proposed neural models is comparable with the performance of the rule-based model for extracting entity phrase from shorter sentences (less than length 15). However, their performance deteriorates for longer sentences. This result fits the characteristics of the training data. As they are trained with shorter sentences, they are unable to generalize the learning patterns for extracting entity phrases from longer sentences. While the best performing LLM-based model achieves an overall 77% F1 score, the rule-based model reaches 81% F1 score. The similar pattern is also reflected for the BLUE score. From the results, we can see that only the fine-tuned GPT-3.5 is able to outperform the rule-based model in entity extraction. This is attributed to its exposure to vast underlying data. However, GPT with zero-shot setup is significantly underperforming for entity extraction as well.

5.3 Discussion

Here, we can observe from the results presented in Table 3 and Table 4 that the proposed models perform comparable to each other. Overall, they outperform baselines for extracting numerical assertions from shorter sentences. However, the rule-based model outperforms the entity extraction task for longer sentences, although overall performance remains inferior to the neural models. We also observe from the results that fine-tuned GPT model significantly outperforms the GPT-4 with zero-shot set up. While fine-tuned GPT 3.5 built on a LLM

Table 4: Performance of models on extraction of entity phrases

Performance for Sentence Length $\leq$ 15)				
Models	**Precision**	**Recall**	**F1 Score**	**BLUE Score**
GPT 4	0.70	0.86	0.77	0.69
Fine-tuned GPT 3.5	**0.87**	0.85	0.86	0.82
Encoder-decode Arch.	0.77	0.87	0.82	0.81
Rule Based Model	**0.87**	0.91	0.89	0.81
Tag-Aware Joint Extraction	**0.87**	0.93	**0.90**	**0.81**
Attribute-Aware Joint Extraction	0.85	**0.96**	**0.90**	0.77
Performance for Sentence Length $>$ 15				
GPT 4	0.59	0.59	0.59	0.50
Fine-tuned GPT 3.5	0.72	0.66	0.69	0.59
Encoder-decode Arch.	0.58	0.61	0.59	0.49
Rule Based Model	**0.79**	**0.74**	**0.76**	**0.65**
Tag-Aware Joint Method	0.66	0.65	0.66	0.54
Attribute-Aware Joint Method	0.63	0.62	0.62	0.48

Table 5: Anecdotal Examples of Extracted Numerical Assertions $\langle$E, M,Q$\rangle$

The bridge carries 60 to 70 percent of commercial truck traffic in the region.			
Methods	Entity	Measuring attribute	Quantity
GPT4	The bridge	share of regional commercial truck traffic carried	60 to 70%
fine-tuned GPT3.5	bridge	commercial truck traffic share	60 to 70%
Rule-based	bridge	truck traffic carrier	60 to 70%
Attribute-aware	bridge	carries commercial truck traffic	60 to 70%
The same poll, 800 Hispanic voters in Florida, had 35 % of non-Cuban Hispanics supporting Mr. Bush, 59 % Mr. Kerry and 6 % undecided or supporting Mr. Nader.			
Methods	Entity	Measuring attribute	Quantity
GPT4	Non-Cuban Hispanics	support for Mr. Bush	35%
fine-tuned GPT3.5	Mr.	Non-Cuban Hispanics supporting Mr. Bush	35%
Rule-based	The Same Poll	non-cuban hispanics supporting	35%
Attribute-aware	Florida	Poll voters	35%

with $\approx$ 175 billion parameters, we consider a comparatively smaller LLM with 108 million parameters only. Our proposed models are trained with only $\approx$ 280 million parameters on a relatively small data set with 10K samples. With these smaller models, we outperform fine-tuned GPT in most cases.

Table 5 shows anecdotal examples for extracting numerical triples from different models. We can see that GPT-4 generates out-of-context tokens as measuring attributes for the first example, which is contributed to its poor performance, reflected in the BLUE and F1 score in Table 3. Fine-tuning GPT brings improvement in this aspect. For the example with longer sentence, none of the methods generates appropriate measuring attribute. Consequently, the performance of entity extraction is affected. Our proposed model able to understand that 35% is about poll voter in Florida, which is partially correct. Here, GPT models tends to capture factual content rather than the measuring attribute. As reflected in the performance scores, all models perform well in extracting entity correctly for shorter sentences.

6 Conclusion

Extracting numerical assertions in the form of $\langle$ entity, measurement attribute, quantity$\rangle$ can enhance the coverage of quantitative facts in knowledge graphs, as well as the understanding of quantity in text. This work presents a rule-based extraction pipeline built on BONIE to generate numerical triples from text. While the extractor demonstrates high precision in identifying entities and quantities, its performance in extracting implicit or context-dependent measuring attributes remains limited. Addressing this limitation, we propose two LLM-based neural models Attribute-aware and Tag-aware Joint extractor that are trained using the dataset generated by the rule-based model. We consider BERT as the underlying LLM, which allows us to create a smaller model with $\approx$ 280 parameters, compared to the dominant general purpose LLM. From the evaluation, we observe that the integration of additional information flow from the encoder to the decoder stack helps to learn a better model that outperforms all baselines for extracting numerical assertions from shorter sentences.

Limitations

Measuring attribute extraction in open-domain settings remains underexplored, and no benchmark datasets exist for numerical assertions. To address this, we leveraged BONIE and designed a rule-based approach to generate a pseudo-labeled dataset for numerical assertions. However, the limitation of BONIE in extracting measuring attribute from longer sentences adversely impacted the quality of measuring attribute annotations, reducing overall extraction quality. This, in turn, restricted the proposed neural model's exposure to various syntactic patterns, limiting its ability to generalize to longer or more complex inputs. Consequently, models often capture only partial measuring attributes and may fail to correctly associate the corresponding entity for longer sentences.

Another limitation of our work stems from the architectural choices in our neural model. We observed that the rule-based approach achieved a high precision for extracting the entity phrase. And therefore, we prioritized learning robust models that can capture the representation of measuring attributes efficiently. Although we investigated the joint extraction of entity and measuring attribute, our design choice remains simple for entity extraction, using sequence tagging over fine-tuned pre-trained language models. This design, combined with training primarily on shorter sentences, restricts generalzibility of the model and affects performance of entity extraction for longer sentences. This further affects the extraction of measuring attribute as both tasks are linked. However, a combined approach that leverages the strengths of both methods remains a promising direction for future work.

References

Omar Alonso and Thibault Sellam. 2018. Quantitative information extraction from social data. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR 2018, Ann Arbor, MI, USA, July 08-12, 2018*, pages 1005–1008. ACM.

Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M. Dai, Anja Hauth, Katie Millican, David Silver, Slav Petrov, Melvin Johnson, Ioannis Antonoglou, Julian Schrittwieser, Amelia Glaese, Jilin Chen, Emily Pitler, Timothy P. Lillicrap, and 33 others. 2023. Gemini: A family of highly capable multimodal models. *CoRR*, abs/2312.11805.

Janara Christensen, Mausam, Stephen Soderland, and Oren Etzioni. 2011. An analysis of open information extraction based on semantic role labeling. In *Proceedings of the Sixth International Conference on Knowledge Capture*, K-CAP '11, page 113–120, New York, NY, USA. Association for Computing Machinery.

Lei Cui, Furu Wei, and Ming Zhou. 2018. Neural open information extraction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 2: Short Papers*, pages 407–413. Association for Computational Linguistics.

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.

Oren Etzioni, Michael Cafarella, Doug Downey, Stanley Kok, Ana-Maria Popescu, Tal Shaked, Stephen Soderland, Daniel S Weld, and Alexander Yates. 2004. Web-scale information extraction in know-itall: (preliminary results). In *In Proceedings of the 13th International Conference on World Wide Web (WWW '04),*, pages 100–110. ACM.

Oren Etzioni, Anthony Fader, Janara Christensen, Stephen Soderland, and 1 others. 2011. Open information extraction: The second generation. In *Twenty-Second International Joint Conference on Artificial Intelligence*.

Vinh Thinh Ho, Yusra Ibrahim, Koninika Pal, Klaus Berberich, and Gerhard Weikum. 2019. Qsearch: Answering quantity queries from text. In *The Semantic Web - ISWC 2019 - 18th International Semantic Web Conference, Auckland, New Zealand, October 26-30, 2019, Proceedings, Part I*, volume 11778 of *Lecture Notes in Computer Science*, pages 237–257. Springer.

Keshav Kolluru, Vaibhav Adlakha, Samarth Aggarwal, Mausam, and Soumen Chakrabarti. 2020a. Openie6: Iterative grid labeling and coordination analysis for open information extraction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 3748–3761. Association for Computational Linguistics.

Keshav Kolluru, Samarth Aggarwal, Vipul Rathore, Mausam, and Soumen Chakrabarti. 2020b. Imojie: Iterative memory-based joint open information extraction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 5871–5886. Association for Computational Linguistics.

Pengfei Li, Kezhi Mao, Xuefeng Yang, and Qi Li. 2019. Improving relation extraction with knowledge-attention. 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 2019, Hong Kong, China, November 3-7, 2019*, pages 229–239. Association for Computational Linguistics.

Aman Madaan, Ashish R. Mittal, Mausam, Ganesh Ramakrishnan, and Sunita Sarawagi. 2016. Numerical relation extraction with minimal supervision. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA*, pages 2764–2771. AAAI Press.

Mausam. 2016. Open information extraction systems and downstream applications. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*, pages 4074–4077. IJCAI/AAAI Press.

Mausam, Michael Schmitz, Stephen Soderland, Robert Bart, and Oren Etzioni. 2012. Open language learning for information extraction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 523–534, Jeju Island, Korea. Association for Computational Linguistics.

Niels Mündler. 2022. quantulum3. `https://pypi.org/project/quantulum3/`.

OpenAI. 2023. GPT-4 technical report. *CoRR*, abs/2303.08774.

Harinder Pal and Mausam. 2016. Demonyms and compound relational nouns in nominal open IE. In *Proceedings of the 5th Workshop on Automated Knowledge Base Construction*, pages 35–39, San Diego, CA. Association for Computational Linguistics.

Kevin Pei, Ishan Jindal, Kevin Chen-Chuan Chang, ChengXiang Zhai, and Yunyao Li. 2023. When to use what: An in-depth comparative empirical analysis of openie systems for downstream applications. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 929–949. Association for Computational Linguistics.

Subhro Roy, Tim Vieira, and Dan Roth. 2015. Reasoning about quantities in natural language. *Trans. Assoc. Comput. Linguistics*, 3:1–13.

Swarnadeep Saha, Harinder Pal, and Mausam. 2017. Bootstrapping for numerical open IE. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 317–323, Vancouver, Canada. Association for Computational Linguistics.

Yuanhe Tian, Guimin Chen, Yan Song, and Xiang Wan. 2021. Dependency-driven relation extraction with attentive graph convolutional networks. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4458–4471, Online. Association for Computational Linguistics.

Michael Vasilkovsky, Anton Alekseev, Valentin Malykh, Ilya Shenbin, Elena Tutubalina, Dmitriy Salikhov, Mikhail Stepnov, Andrey Chertok, and Sergey I. Nikolenko. 2022. Detie: Multilingual open information extraction inspired by object detection. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*, pages 11412–11420. AAAI Press.

Shanchan Wu and Yifan He. 2019. Enriching pre-trained language model with entity information for relation classification. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2019, Beijing, China, November 3-7, 2019*, pages 2361–2364. ACM.

Alexander Yates, Michele Banko, Matthew Broadhead, Michael Cafarella, Oren Etzioni, and Stephen Soderland. 2007. TextRunner: Open information extraction on the web. In *Proceedings of Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT)*, pages 25–26, Rochester, New York, USA. Association for Computational Linguistics.

Dongxu Zhang and Dong Wang. 2015. Relation classification via recurrent neural network. *Preprint*, arXiv:1508.01006.

Pengyu Zhang. 2022. A numerical fact extraction method for chinese text. In *7th IEEE International Conference on Smart Cloud, SmartCloud 2022, Shanghai, China, October 8-10, 2022*, pages 97–103. IEEE.

Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D. Manning. 2017. Position-aware attention and supervised data improve slot filling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 35–45, Copenhagen, Denmark. Association for Computational Linguistics.

Xiaoyan Zhao, Yang Deng, Min Yang, Lingzhi Wang, Rui Zhang, Hong Cheng, Wai Lam, Ying Shen, and Ruifeng Xu. 2024. A comprehensive survey on relation extraction: Recent advances and new frontiers. *ACM Comput. Surv.*, 56(11):293:1–293:39.