NER-MQMRC: Formulating Named Entity Recognition as Multi Question Machine Reading Comprehension

Anubhav Shrimal[1] **Avi Jain**[2*] **Kartik Mehta**[3†] **Promod Yenigalla**[1]

[1]Retail Business Services, Amazon
[2]Product Graph, Amazon
[3]Amazon Search, Amazon
{shrimaa, jainav, kartim, promy}@amazon.com

Abstract

NER has been traditionally formulated as a sequence labeling task. However, there has been recent trend in posing NER as a machine reading comprehension task (Wang et al., 2020; Mengge et al., 2020), where entity name (or other information) is considered as a question, text as the context and entity value in text as answer snippet. These works consider MRC based on a single question (entity) at a time. We propose posing NER as a multi-question MRC task, where multiple questions (one question per entity) are considered at the same time for a single text. We propose a novel BERT-based multi-question MRC (NER-MQMRC) architecture for this formulation. NER-MQMRC architecture considers all entities as input to BERT for learning token embeddings with self-attention and leverages BERT-based entity representation for further improving these token embeddings for NER task. Evaluation on three NER datasets show that our proposed architecture leads to average 2.5 times faster training and 2.3 times faster inference as compared to NER-SQMRC framework based models by considering all entities together in a single pass. Further, we show that our model performance does not degrade compared to single-question based MRC (NER-SQMRC) (Devlin et al., 2019) leading to F1 gain of +0.41%, +0.32% and +0.27% for AE-Pub, Ecommerce5PT and Twitter datasets respectively. We propose this architecture primarily to solve large scale e-commerce attribute (or entity) extraction from unstructured text of a magnitude of 50k+ attributes to be extracted on a scalable production environment with high performance and optimised training and inference runtimes.

1 Introduction

Named Entity Recognition (NER) is the task of locating and classifying entities mentioned in unstructured text into predefined categories such as names of people, organizations and locations. It is a crucial component of many applications, such as web search, relation extraction (Yu et al., 2019) and e-commerce attribute extraction (Zheng et al., 2018; Mehta et al., 2021). Traditionally, NER has been posed as a sequence labeling task (Ma and Hovy, 2016; Zheng et al., 2018; Devlin et al., 2019) where each token is assigned a single tag class. We term these sequence labeling approaches as NER-SL. Recently, there has been interest in posing NER as a machine reading comprehension task (Wang et al., 2020; Mengge et al., 2020; Xu et al., 2019). Specifically, NER is posed as a question answering problem, where text is considered context, entity name (or some variant) is considered question and entity value mentioned in text is considered as answer snippet. We term these approaches as Single Question Machine Reading Comprehension (NER-SQMRC) as they involve asking a single question (or entity) at a time. We argue that both NER-SL and NER-SQMRC have their merits and demerits, e.g. NER-SQMRC incorporates entity name for better representation and can be easily extended to new entities without re-training and NER-SL requires single scoring pass for extracting all entities from a given text. We pose NER as a multi-question MRC problem, where multiple questions (one question per entity) are asked at the same time and propose a novel architecture (NER-MQMRC) for this formulation. We summarize the merits and demerits of these three formulations in Table 1 considering below factors:

- **Entity scaling**: Ability to scale for new entities without retraining.
- **Multi-entity scoring**: Ability to extract all entities from a given text in a single forward pass.
- **Faster runtime**: Extracting multiple entities together in a single pass leads to faster training and inference as compared to considering single entity in a pass.

*work done as part of Retail Business Services, Amazon
†work done as part of India Machine Learning, Amazon

Proceedings of NAACL-HLT 2022: Industry Track Papers, pages 230 - 238

- **Using entity information**: Leveraging entity information (such as entity name) for learning better representations.

Property	NER-SL	NER-SQMRC	NER-MQMRC
Entity scaling	✗	✔	✔
Multi-entity Scoring	✔	✗	✔
Faster runtime	✔	✗	✔
Entity information	✗	✔	✔

Table 1: Comparing different attribute extraction approaches based on various factors.

As summarized in Table 1, our proposed NER-MQMRC architecture combines the best of NER-SL and NER-SQMRC. NER-MQMRC considers extraction of multiple entities based on multiple questions on same text, and is novel in three ways - 1) Token representations are learnt to incorporate information of all the entities, unlike using single entity as in (Wang et al., 2020; Mengge et al., 2020). 2) We introduce leveraging BERT-based entity representations for further improving token representations for NER task. 3) Our architecture leads to faster training and inference. E.g. scoring of five entities can be done using a single forward pass with our NER-MQMRC as compared to five passes required earlier with NER-SQMRC based models (Devlin et al., 2019; Wang et al., 2020; Mengge et al., 2020). Experiments on three NER datasets establish the effectiveness of NER-MQMRC architecture. NER-MQMRC achieves 2.5x faster training and 2.3x faster inference as compared to single question based MRC (NER-SQMRC) framework based models by considering multiple entities together in training and inference. Further, we show performance boost over SOTA NER-SQMRC (Devlin et al., 2019), obtaining +0.41%, +0.32% and +0.27% F1 improvements for AE-Pub, Ecommerce5PT and Twitter datasets respectively. Rest of the paper is organized as follows. We describe our proposed NER-MQMRC architecture in Section 2. We discuss our experimental setup in Section 3 followed by results in Section 4. We discuss the industry impact of our work in Section 5 and summarize the paper in Section 6.

2 NER as a Multi-Question MRC task

2.1 Problem definition and dataset construction

Given an input sequence $X = \{x_1, x_2, ..., x_n\}$, where n denotes the length of the sequence, the objective in NER task is to find and label tokens in X that represent entity $y \in Y$, where Y is a predefined list of all possible entities (e.g., BRAND, COLOR, etc). Under the NER-SQMRC framework, the model is given a question q_i asking about i^{th} entity and the model has to extract a text span x_{start_i,end_i} from X which are tokens corresponding to the i^{th} entity. For NER-MQMRC framework, the model is given a list of k questions $Q = \{q_1, q_2, ..., q_k\}$ and the model has to extract the text spans $\{(x_{start_1,end_1}), (x_{start_2,end_2}), ..., (x_{start_k,end_k})\}$ from X corresponding to each of the k entities. We use BERT for Question Answering (Devlin et al., 2019) as our NER-SQMRC baseline implementation (refer Appendix A.1).

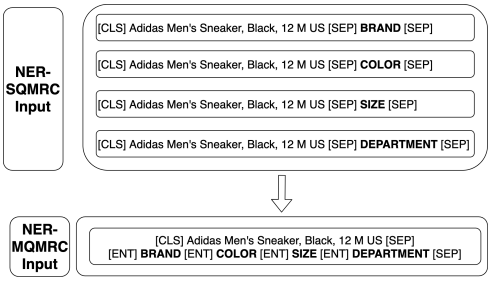

Figure 1: Data Input format for NER-SQMRC and NER-MQMRC model architectures.

Figure 1 shows data input format for both NER-SQMRC and NER-MQMRC. Similar to conventional Question Answering, training data for NER-SQMRC consists of (text, single-entity-question, entity spans from text) triplets. For a dataset with k entities, training data consists of k samples for each text, each sample having question for one entity. However, for NER-MQMRC, training data consists of a single sample for each text, having k questions (one question per entity). Hence, NER-SQMRC formulation requires dealing with larger size training data (k times more samples) with same information as compared to NER-SL and NER-MQMRC. Similarly, during inference, NER-SQMRC requires performing k evaluations for the same text to get text span for each entity, whereas NER-MQMRC requires only a single evaluation for all entities.

2.2 Model Details

Figure 2 shows our proposed NER-MQMRC architecture. We build NER-MQMRC on top of BERT architecture (Devlin et al., 2019) by customizing BERT input and modifying the output layer as described in this section.

2.2.1 NER-MQMRC input

BERT has been trained to take a pair of sentences separated by a special token *[SEP]* as input, and use E_A and E_B segment embeddings respectively for tokens of each sentence. For NER-MQMRC, we concatenate the input text and questions of all entities separated by *[SEP]* (refer Figure 1). Questions of each entity are further separated by a special token *[ENT]*, which we add to the BERT vocabulary. We use E_A segment embeddings for input text and E_B segment embeddings for all question tokens. Output embedding learned corresponding to each *[ENT]* token is considered as embedding representation for the entity adjacent to that *[ENT]* token.

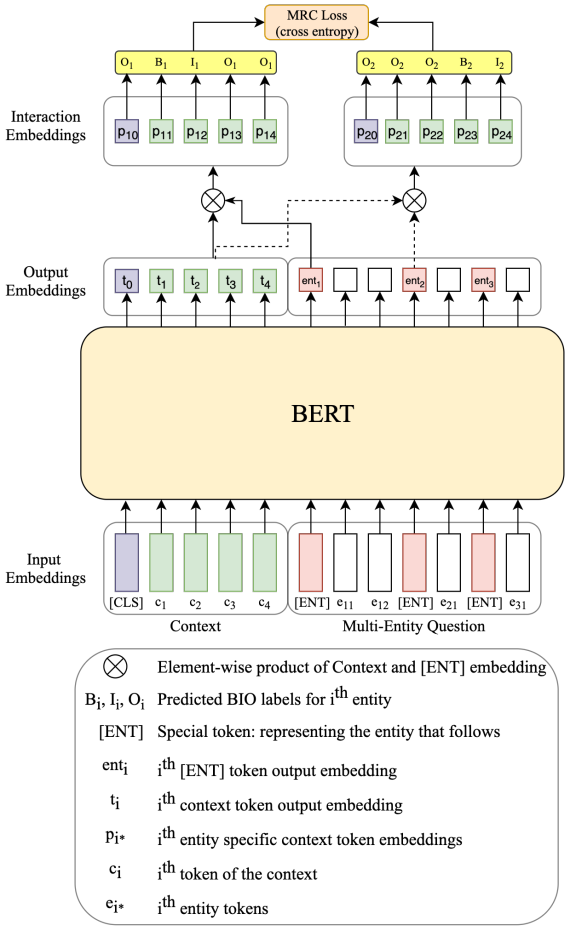

Figure 2: NER-MQMRC model architecture.

2.2.2 Entity specific representation and span selection

As discussed, the i^{th} *[ENT]* token output embedding (ent_i) represents the i^{th} entity in the question. We hypothesize that using ent_i to attend to the context token's output embeddings, $T = \{t_1, t_2, ..., t_n\}$, will help the model find the answer span for entity i. We use entity embeddings to transform the common context representations (T) to entity specific token representations. We consider extraction of each entity as a separate task and use element-wise product of token and entity embeddings to obtain entity specific representations for each token (refer Figure 2). More formally, we perform an element-wise product of token embeddings T with ent_i to get i^{th} entity specific token representations $P_i = \{p_{i1}, p_{i2}, ..., p_{in}\}$.

These entity specific representations are then fed into a separate token-level dense layer, W_{bio}, to get the BIO format label prediction for each token w.r.t. the entity as shown in equation 1, where t_j represents embedding for j^{th} token and ent_i represents embedding for i^{th} entity. Examples with no entity mention are modelled by setting the label for *[CLS]* token as B tag for that entity. For each token and entity pair, loss is calculated using cross entropy loss (L^{ce}) between predicted and actual label. For each sample, total loss, L_{total} (refer equation 2), is average (with equal weightage) of loss for all k entities and n tokens pairs.

$$label_j = \arg\max(softmax(W_{bio}(t_j \odot ent_i))) \quad (1)$$

$$L_{total} = \frac{1}{k \cdot n} \sum_{i=1}^{k} \sum_{j=1}^{n} L_{i,j}^{ce} \quad (2)$$

2.3 Discussion

Our proposed formulation is generic and can be used with other pre-trained architectures (such as XLNET, RoBERTa) instead of BERT for feature extraction. In recent years, there has been incremental advancements to the MRC framework such as the use of knowledge distillation loss as a regularizer and no-answer loss (Wang et al., 2020) to achieve better performance than (Devlin et al., 2019); NER-MQMRC framework can also easily integrate such ideas to get better performance and we keep this to be explored as a future work since in this paper we want to show the effectiveness of NER-MQMRC framework over NER-SQMRC framework with similar setup for both the frameworks. We use BIO label prediction to allow multiple value predictions for an entity from the text, though we also experimented with single (start, end) span index prediction as output labels similar to (Wang et al.,

Dataset	Train Data SQMRC	MQMRC	Reduction(%)	Test Data SQMRC	MQMRC	Reduction(%)
Ecommerce5PT	981,076	290,698	**70.37**	32,062	4,967	**84.51**
AE-Pub	88,460	39,888	**54.91**	22,005	17,393	**20.96**
Twitter	11,997	3,999	**66.67**	9,768	3,256	**66.67**

Table 2: Reduction in dataset size due to single-entity to multi-entity question transformation.

2020) but has the limitation of predicting only a single answer span.

3 Experimental Setup

3.1 Datasets

Experiments were performed on three NER datasets described below.

AE-Pub (Xu et al., 2019) is a dataset for E-commerce attribute extraction collected from AliExpress Sports & Entertainment category. This dataset is designed to pose E-commerce attribute extraction as a question answering problem and contains over 110k triplets (text, attribute, value) and 2.7k unique attributes. Even though the number of attributes is large, any given text in the dataset has no more than 13 attributes. Train and test dataset is created in an automated manner using distant supervision.

Ecommerce5PT is a 33 attributes (size, material, color, etc.) dataset extracted from five different product types from Amazon catalogue. The train data is constructed in a similar way as AE-Pub using distant supervision. The train data quality is improved using automated gazetteer and matching heuristics (refer Appendix A.2). Unlike AE-pub, test data is constructed with manual audit, thus leading to better quality test data.

Twitter (Zhang et al., 2018) is an English NER dataset based on tweets. We use the setup similar to (Mengge et al., 2020), using textual information queries (refer Appendix A.5) and making entity detection on PER, LOC and ORG.

3.1.1 Datasets transformation

As discussed earlier, NER-MQMRC leads to reduced train and test data size as compared to NER-SQMRC (Table 2). We observe a median of 3, 2 and 3 entities per question in training data of Ecommerce5PT, AE-Pub and Twitter datasets respectively, leading to similar data reduction for NER-MQMRC training. Appendix A.4 elaborates on the distribution of entities per question for NER-MQMRC for each of these datasets. For fair comparison, one should use all entities of a sample while evaluating NER-SL, NER-SQMRC and NER-MQMRC approaches. We use this setup for Ecommerce5PT and Twitter datasets. However, AE-Pub dataset contains only few entities of each sample. We follow setup used in (Xu et al., 2019) for AE-Pub evaluation.

3.2 Experiments

In this section we detail the various experiments to evaluate our proposed solution, NER-MQMRC, on aspects such as operational performance (training and inference runtime), NER task, limited data setting (few shot) and NER-MQMRC model specific analysis.

Training and Inference Runtime: We compare how much time does NER-SQMRC and NER-MQMRC take to do one pass over the complete train data (1 epoch) as well as for inference on complete test data. For a fair comparison, the models are run on the same machine and under the same conditions.

Named Entity Recognition: We evaluate models for the task of extracting entities from a given text. For NER-SL models (Mehta et al., 2021; Ma and Hovy, 2016), input is a text in which tokens are to be tagged with entity BIO labels (B-PER, I-LOC, etc.). For NER-SQMRC models (Devlin et al., 2019; Wang et al., 2020; Xu et al., 2019), input is a text and a corresponding single entity question, whereas, for our proposed NER-MQMRC models, input is a text and a multi-entity question (section 2.1). The output for each model (NER-SL, NER-SQMRC and NER-MQMRC) are BIO labels for each token in the text. We use micro average precision (P), recall (R) and F_1 as evaluation metrics and use Exact Match criteria (Rajpurkar et al., 2016) to compute the scores.

Few-shot Learning: We analyze the performance as the number of data samples seen during training are reduced. We perform this analysis using Ecommerce5PT dataset and compare with Multi-task NER architecture (Mehta et al., 2021).

Context-Entity Interaction: Element-wise product operation is applied over entity embedding and token output embeddings to get entity specific token embeddings. As the operation performed is important to filter information, in this experiment we explore the effects of using different operations other than using element-wise product.

Impact of entity ordering: We evaluate the impact on model performance due to the order in which en-

Ecommerce5PT			
methods	P(%)	R(%)	F1(%)
Multi-task NER (Mehta et al., 2021) (single model)	**91.62**	62.47	74.29
Multi-task NER (Mehta et al., 2021) (5 model ensemble)*	88.90	77.20	82.60
BERT-Tagger (Devlin et al., 2019)	88.43	77.51	82.61
NER-SQMRC (Devlin et al., 2019)	87.92	81.18	84.42
NER-MQMRC	87.52	**82.14**	**84.74**
AE-Pub			
methods	P(%)	R(%)	F1(%)
SUOpenTag (Xu et al., 2019)	79.85	70.57	74.92
AVEQA (Wang et al., 2020)	86.11	**83.94**	**85.01**
NER-SQMRC (Devlin et al., 2019)	85.08	83.19	84.13
NER-MQMRC	**86.18**	82.97	84.54
Twitter			
methods	P(%)	R(%)	F1(%)
BiLSTM-CRF (Ma and Hovy, 2016)	-	-	65.32
CoFEE-MRC (Mengge et al., 2020)	75.89	71.93	73.86
NER-SQMRC (Devlin et al., 2019)	**80.37**	76.90	78.59
NER-MQMRC	77.79	**79.96**	**78.86**

∗ Five individual models were trained and evaluated, one for each product type
AVEQA (Wang et al., 2020) uses no-answer and distillation loss as regularizers

Table 3: Performance comparison on various NER datasets.

tities are mentioned in a question as NER-MQMRC is formulated as a multi-entity question.

4 Results

4.1 Operational Performance – training and inference runtime

Figure 3 shows the relative training and inference time of NER-MQMRC and NER-SQMRC on all three datasets. We observe that NER-MQMRC leads to an average 2.5 times faster training and 2.3 times faster inference due to performing single forward pass for all entities, as compared to NER-SQMRC which requires a separate forward pass for each entity. The runtime improvement depends on how many entities are grouped together in the dataset for each text. NER-MQMRC inference runtime on AE-Pub is only 5% faster than NER-SQMRC as only 20.96% reduction happened in test dataset size after data transformation (Table 2).

4.2 NER Task Performance

Table 3 shows comparison of our proposed model with baselines on multiple NER datasets. Based on evaluation on three NER datasets, our proposed

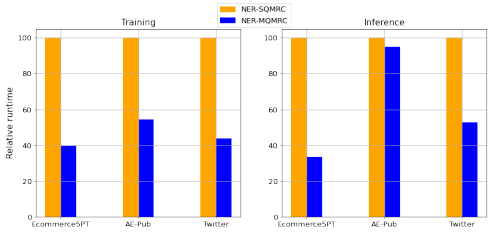

Figure 3: Comparison of operational metrics.

model outperforms NER-SQMRC (Devlin et al., 2019) achieving F1 gain of +0.41%, +0.32% and +0.27% for AE-Pub, Ecommerce5PT and Twitter datasets respectively. A single NER-MQMRC model outperforms ensemble of five Multi-task NER models (one for each product type) by +2.14% F1 and helps in avoiding model proliferation by having a single model instead of a different model for each product type for Ecommerce5PT dataset. NER-MQMRC outperforms BERT-Tagger (Devlin et al., 2019) by +2.13% which uses BERT for NER as a tagging task (NER-SL). For AE-Pub, NER-MQMRC has 0.47% lower F1 compared to AVEQA (Wang et al., 2020), which is due to the additional No-answer and Distillation loss components in AVEQA. Note that NER-MQMRC is agile and such modules can be easily integrated to it as well.

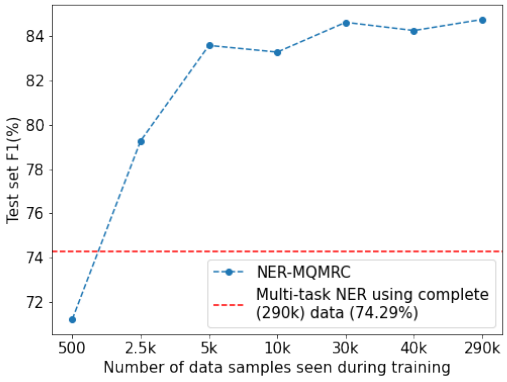

Figure 4: Performance with less training data on Ecommerce5PT.

4.3 Evaluation in limited data setting – Few-shot Learning

Figure 4 shows the performance of NER-MQMRC with lesser data availability during training. NER-MQMRC is able to perform better than Multi-Task NER model trained on complete Ecommerce5PT data (290k samples) with as low as 2.5k samples

Operation Name	Formula	P(%)	R(%)	F1(%)
layer_sum	$P_i = W_1(T) + W_2(ent_i)$	79.07	11.46	20.02
difference	$P_i = T - ent_i$	85.99	20.70	33.36
layer_product_relu	$P_i = relu(W_1(T)) * relu(W_2(ent_i))$	74.80	79.57	77.11
layer_product_tanh	$P_i = tanh(W_1(T)) * tanh(W_2(ent_i))$	76.68	78.49	77.58
max	$P_i = max(T, ent_i)$	76.87	**80.79**	78.78
element-wise product	$P_i = T * ent_i$	77.79	79.96	78.86
layer_product	$P_i = W_1(T) * W_2(ent_i)$	**78.87**	79.66	**79.26**

W_1, W_2 are linear weight matrices
T is the context vector of shape (n, dim) where n is the context length
ent_i is the entity vector of shape $(dim,)$ for the ith entity
P_i is the ith entity specific context vector of shape (n, dim)
$*, +, -$ and max are element-wise product, sum, difference and max operations respectively

Table 4: Effect of different operations to attend context vectors using entity vector.

during training. NER-MQMRC is able to perform even with few samples for training because of the natural language understanding a pre-trained BERT model possesses. The performance further increases with increase in dataset size. For Multi-task NER model we observed the F1 further dropped from 74.29% to 54.49% when trained with 40k data samples.

4.4 NER-MQMRC Specific Experiments

4.4.1 Context Entity Interaction Operations

We experimented with a list of different operations to get better entity specific context embeddings on Twitter dataset. As shown in Table 4, *layer_product* operation performed the best with 79.26% F1. Operations such as element-wise *sum* and *difference* performed poorly in generating good quality entity specific context embeddings because they did not amplify the context vector features by large magnitudes which helps the classification layer better differentiate whereas *product* operation amplified the feature magnitudes.

4.4.2 Effects of entity ordering in a question

We observe that keeping the same ordering of entities in a question while training, leads to deterioration in F1 if the entities are then shuffled during inference (-12.33% on average). This is likely due to model giving more weightage to relative entity position while learning the entity representations and not focusing on the entity name (or entity question). Shuffling the order of entities during training alleviates this issue and leads to robust results for any order of entities during evaluation.

5 Industry Impact

Cost saving: Our production pipeline uses AWS p2.8xlarge compute instance for model training which costs $7.2/hour. Training a single NER-SQMRC model takes 17 hours whereas our proposed NER-MQMRC model takes 7 hours which saves $72 per model training i.e. reducing the model training cost by an average of 58.82%. Training multiple such models leads to large cost savings for production systems.
Faster model runtime: Due to the faster training and inference capabilities of NER-MQMRC, our production systems are deployed faster and are able to serve 2.3 times more inference requests per minute improving the model throughput.
Model proliferation reduction: The NER-SL based production systems need to deploy multiple models as they are not able to perform at scale with the increase in the number of attributes in the e-commerce catalogue due to the increase in output label space. NER-MQMRC alleviates this issue as the output label space remains constant (3 for BIO labels) and a single model can be trained for 50k+ number of attributes.
Better performance: From our experiments we show that NER-MQMRC performs better than NER-SL and NER-SQMRC framework models.

6 Conclusion

In this paper, we formulated NER as a multi question MRC task (NER-MQMRC). Experimental evaluation on three NER datasets shows that our proposed NER-MQMRC model handles multiple entities together and leads to faster training and inference as compared to single question MRC for-

mulation and improves performance over SOTA NER-SQMRC model (Devlin et al., 2019), establishing the effectiveness of our proposed model.

References

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.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1064–1074, Berlin, Germany. Association for Computational Linguistics.

Kartik Mehta, Ioana Oprea, and Nikhil Rasiwasia. 2021. LATEX-numeric: Language agnostic text attribute extraction for numeric attributes. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Industry Papers*, pages 272–279, Online. Association for Computational Linguistics.

Xue Mengge, Bowen Yu, Zhenyu Zhang, Tingwen Liu, Yue Zhang, and Bin Wang. 2020. Coarse-to-Fine Pre-training for Named Entity Recognition. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6345–6354, Online. Association for Computational Linguistics.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Qifan Wang, Li Yang, Bhargav Kanagal, Sumit Sanghai, D. Sivakumar, Bin Shu, Zac Yu, and Jon Elsas. 2020. Learning to extract attribute value from product via question answering: A multi-task approach. In *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020*, pages 47–55. ACM.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

Huimin Xu, Wenting Wang, Xin Mao, Xinyu Jiang, and Man Lan. 2019. Scaling up open tagging from tens to thousands: Comprehension empowered attribute value extraction from product title. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5214–5223, Florence, Italy. Association for Computational Linguistics.

Bowen Yu, Zhenyu Zhang, Tingwen Liu, Bin Wang, Sujian Li, and Quangang Li. 2019. Beyond word attention: Using segment attention in neural relation extraction. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pages 5401–5407. ijcai.org.

Qi Zhang, Jinlan Fu, Xiaoyu Liu, and Xuanjing Huang. 2018. Adaptive co-attention network for named entity recognition in tweets. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 5674–5681. AAAI Press.

Guineng Zheng, Subhabrata Mukherjee, Xin Luna Dong, and Feifei Li. 2018. Opentag: Open attribute value extraction from product profiles. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018, London, UK, August 19-23, 2018*, pages 1049–1058. ACM.

A Appendix

A.1 NER as Single Question MRC

Figure 5 shows our baseline NER-SQMRC architecture. We use BERT for Question Answering (Devlin et al., 2019) as our NER-SQMRC baseline implementation. The model is given a question q_i asking about i^{th} entity and the model has to extract a text span x_{start_i,end_i} from X which are tokens corresponding to the i^{th} entity. The question component of the input in NER-SQMRC comprises of a single entity of interest to be extracted. The context token embeddings derived from the forward pass of the BERT model are then used to extract the text span corresponding to the entity from the context. For a text with five entities the NER-SQMRC model will need to perform five forward pass through the model to extract the text spans for each of the five entities.

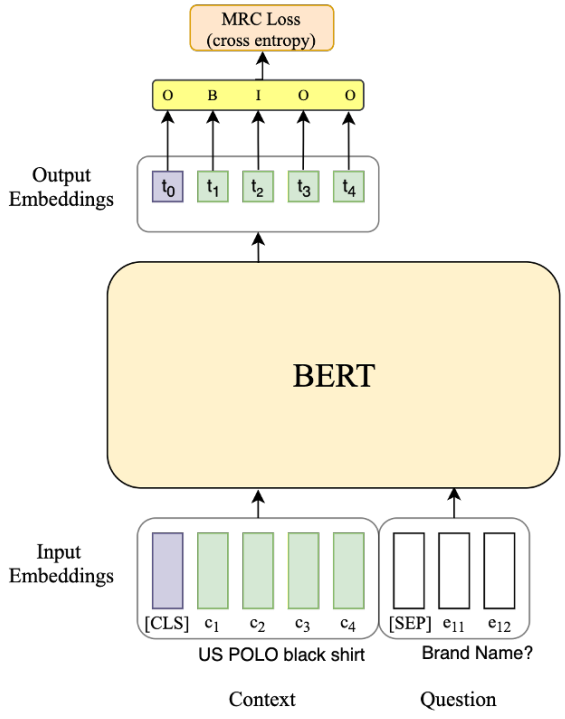

Figure 5: NER-SQMRC model architecture.

A.2 Ecommerce5PT training data generation

Catalogue attribute values can be noisy (e.g. having junk value or missing value) and leads to noisy training annotations with distant supervision. In this section we explain the strategies employed to create better quality training data for Ecommerce5PT dataset.

A.2.1 Automated Gazetteer

Using gazetteers in distant supervision can improve the quality of training annotations (especially for attributes which have limited set of valid values). As part of the data tagging step, the catalogue backend values for an attribute are read to create the gazetteer values using the most frequently occuring attribute values. Elbow method is used to determine the threshold for selecting values for the gazetteer. The training data is then created leveraging the backend attribute values and gazetteer values in distant supervision.

A.2.2 Other Heuristics

The backend catalogue value sometimes contains a different variation of the attribute value than what is present in the context. For example, context is "US Polo t-shirt for Men" whereas the backend value for the attribute *target-audience* is "Man". Such cases will not be tagged using exact match in distant supervision. Custom heuristics such as pluralizing the text (Men, Mens, Men's, etc.), removing or adding "s", lower casing the text and normalizing attributes such as converting "XXXXL" to "4XL" for *size* attribute are added to improve the training data quality.

A.3 Implementation Details

In this section we discuss the dataset creation and model training hyper-parameters details to replicate our results.

During training, we explicitly add no answers for entities that do not have a span in a given text to make the model learn to predict *[CLS]* if no valid answer is present for an entity. We do not make any additions to AE-Pub since it already has no answers added for certain entities. For Ecommerce5PT we add 60% no answers at random and for Twitter and CoNLL we add all no answers for each entity that is not present in that text.

For our implementation of NER-SQMRC and NER-MQMRC, we use the transformers library (Wolf et al., 2019). We use variants of pre-trained BERT model for all our experiments. We use *base-cased* variant for Twitter and *base-uncased* variant for AE-Pub and Ecommerce5PT, keeping our evaluation fairly comparable to existing literature. We use the output layer of single (start, end) span index for AE-Pub dataset similar to (Wang et al., 2020) instead of BIO label. Furthermore, we don't do any dataset specific preprocessing or specific hyperparameter tuning. We use batch size of 32, and

a learning rate of 1e-5. We train our models for 20 epochs, choosing the best epoch based on results on the dev set. We make use of AWS compute (ml.p3.8xlarge) instances to run our experiments.

A.4 Entities per question

Figure 6 shows the distribution of number of entities in a question for different datasets. It can be seen from the figure the number of entities in a question greater than 1 are frequent in these datasets which is inefficient for SQMRC type models since they require one forward pass per entity for the same text. We found that Ecommerce5PT and AE-Pub datasets have as many as 12 and 13 attributes for a single text respectively. For Twitter we add all the entities in the question as the dataset has only 3 attributes.

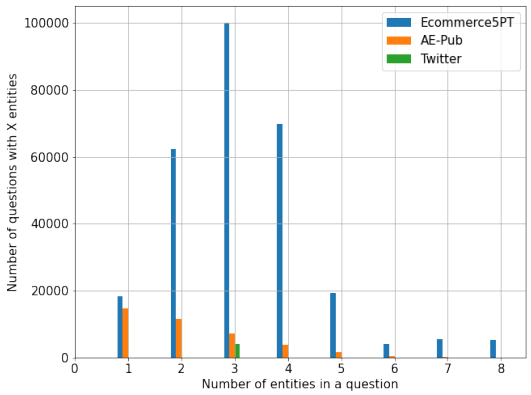

Figure 6: Distribution of number of entities per question in train splits of NER datasets.

Entity Label	Query
PER	People, persons, including fictional
ORG	Companies, agencies, institutions, organizations
LOC	Places, countries, continents, mountain ranges, water bodies

Table 5: Queries used to replace entity label in a question for Twitter.

A.5 Queries

BERT model has natural language understanding capabilities due to large corpus pre-training. This knowledge can be levaraged in MRC to ask better questions. We use a entity description as question instead of entity name in the question so that better representations can be learned by the model. For Twitter we use the language queries in Table 5. For Twitter dataset, only *PER*, *ORG* and *LOC* entity label queries are used because we follow the dataset creation guidelines as stated in (Mengge et al., 2020) where *OTHERS* entity label is ignored.