From Dataset Recycling to Multi-Property Extraction and Beyond

Tomasz Dwojak[1,2], **Michał Pietruszka**[1,3], **Łukasz Borchmann**[1,4], **Jakub Chłędowski**[1,3], **and Filip Graliński**[1,2]

[1]Applica.ai
[2]Faculty of Mathematics and Computer Science, Adam Mickiewicz University in Poznan
[3]Faculty of Mathematics and Computer Science, Jagiellonian University
[4]Institute of Computing Science, Poznan University of Technology

tomasz.dwojak@applica.ai

Abstract

This paper investigates various Transformer architectures on the WikiReading Information Extraction and Machine Reading Comprehension dataset. The proposed dual-source model outperforms the current state-of-the-art by a large margin. Next, we introduce WikiReading Recycled—a newly developed public dataset, and the task of multiple-property extraction. It uses the same data as WikiReading but does not inherit its predecessor's identified disadvantages. In addition, we provide a human-annotated test set with diagnostic subsets for a detailed analysis of model performance.

1 Introduction

The emergence of attention-based models has revolutionized Natural Language Processing (Young et al., 2018). Pretraining these models on large corpora like BookCorpus (Zhu et al., 2015) has been shown to yield a reliable and robust base for downstream tasks. These include Natural Language Inference (Bowman et al., 2015), Question Answering (Rajpurkar et al., 2016), Named Entity Recognition (Yadav and Bethard, 2018; Goyal et al., 2018; Li et al., 2020), and Property Extraction (Hewlett et al., 2016).

The creation of large supervised datasets often comes with trade-offs, such as one between the quality and quantity of data. For instance, the WikiReading dataset (Hewlett et al., 2016) has been created in such a way that WikiData annotations were treated as the expected answers for related Wikipedia articles. However, the above datasets were created separately, and the information content of both sources overlaps only partially. Hence, the resulting dataset may contain noise.

The best models can achieve results better than the human baseline across many NLP datasets such as MSCQAs (Wang et al., 2018), STS-B, QNLI (Raffel et al., 2020), CoLA or MRPC (Wang et al., 2020). However, as a consequence of different kinds of noise in the data, they rarely maximize the score metric (Stanislawek et al., 2019). While current work in NLP is focused on preparing new datasets, we regard recycling the current ones as equally important as creating a new one. Thus, after outperforming previous state-of-the-art on WikiReading, we investigated the dataset's weaknesses and created an entirely new, more challenging Multi-Property Extraction task with improved data splits and a reliable, human-annotated test set.

Contribution. The specific contributions of this work are the following. We analyzed the WikiReading dataset and pointed out its weaknesses. We introduced a Multi-Property Extraction task by creating a new dataset: WikiReading Recycled. Our dataset contains a human-annotated test set, with multiple subsets aimed to benchmark qualities such as generalization on unseen properties. We introduced a Mean-Multi-Property-F_1 score suited for the new Multi-Property Extraction task. We evaluated previously used architectures on both datasets. Furthermore, we showed that pretrained transformer models (Dual-Source RoBERTa and T5) beat all other baselines. The new dataset and all the models mentioned in the present paper were made publicly available on GitHub.[1]

2 Related Work

Early work in relation extraction revolves around problems crafted using distant supervision methods, which are semi-supervised methods that automatically label pools of unlabeled data (Craven and Kumlien, 1999). In contrast, many QA datasets were created through crowd-sourcing, where annotators were asked to formulate questions with

[1]https://github.com/applicaai/multi-property-extraction

Proceedings of the 24th Conference on Computational Natural Language Learning, pages 641–651

https://doi.org/10.18653/v1/P17

Dataset	Task	Input	Output
SNLI	Natural Language Inference	two sentences	relation between the sentences
SQUAD	Question Answering	article, question	answer to the question
WiNER	Named Entity Recognition	article	annotated named entities
WR	Property Extraction	article, property	value of the property
WRR (ours)	Multi-Property Extraction	article, properties	values of the properties

Table 1: Comparison of NLP tasks on text comprehension and information extraction. More differences between WR and WRR were outlined in Table 3.

answers that require knowledge retrieval and information synthesis. One of the most popular QA datasets is Wikipedia-based SQUAD, where an instance consists of a human-formulated question, and an encyclopedic reading passage used to base the answer on (Rajpurkar et al., 2018). Another crowd-sourced dataset that profoundly influenced Natural Language Inference research is SNLI (Bowman et al., 2015)—a three-way semantics-based classification of a relation between two different sentences.

Both SQUAD and SNLI are large-scale Machine Reading Comprehension (MRC) tasks, but they cannot be treated as Property Extraction as defined in Section 3; hence they are not considered in this paper. Similarly, some MRC problems framed in TREC tracks, such as Conversational Assistance or Question Answering, are beyond the scope of this paper (Dalton et al., 2020; Dang et al., 2007).

Hewlett et al. (2016) proposed the WikiReading dataset that consists of a Wikipedia article and related WikiData statement. No additional annotation work was performed, yet the resulting dataset was of presumably high reliability. Nevertheless, we consider an additional human annotation to be desired (Section 4.3). Alongside the dataset, a property extraction task was introduced. The idea behind it is to read an article given a property name and to infer the associated value from the article. The property extraction paradigm is described in detail in Section 3, whereas a brief comparison to related datasets is presented in Table 1.

Initially, the best-performing model used placeholders to allow rewriting out-of-vocabulary words to the output. Next, Choi et al. (2017) presented a reinforcement learning approach that improved results on a challenging subset of the 10% longest articles. This framework was extended by Wang and Jin (2019) with a self-correcting action that removes the inaccurate answer from the answer generation module and continues to read.

Data split	Size	In train	%
Validation set	1,452,591	1,374,820	94.65
Test set	821,409	780,639	95.04

Table 2: The size of WikiReading splits (*Size*) and number of articles leaked from the train set as an absolute value or percentage.

Hewlett et al. (2017) hold the state-of-the-art on WikiReading with their proposition of SWEAR that attends over a sliding window's representations to reduce documents to one vector from which another GRU network generates the answer (Chung et al., 2014). Additionally, they evaluated a strong semi-supervised solution on a randomly sampled 1% subset of WikiReading.

To the best of our knowledge, no authors validated Transformer-based models on WikiReading and pretrained encoders.

3 Property Extraction

Let a *property* denote any query for which a system is expected to return an answer from given text. Examples include *country of citizenship* for a biography provided as an input text, or *architect name* for an article regarding the opening of a new building. Contrary to QA problems, a query is not formulated as a question in natural language but rather as a phrase or keyword. We use the term *value* when referring to a valid answer for the stated query. Some properties have multiple valid answers; thus, multiple values are expected. Examine the case of Johann Sebastian Bach's biography for which property *sister* has eight values. We will refer to any task consisting of a tuple (properties, text) for which values are to be provided as a property extraction task.

The biggest publicly available dataset for property extraction is WikiReading (Hewlett et al., 2016). The dataset combines articles from

Wikipedia with Wikidata information. The dataset is of great value; however, several flaws can be identified. First, more than 95% of articles in the test set appeared in the train set (Table 2). Second, the unjustifiably large size of the test set is a substantial obstacle for running experiments. For instance, it takes 50 hours to process the test set using a Transformer model such as $T5_{SMALL}$ on a single NVidia V100 GPU. Finally, WikiReading assumes that every value in the test set can be determined on the basis of a given article. As shown later, this is not the case for 28% of values.

3.1 Towards Multi-Property Extraction

In the Multi-Property Extraction (MPE) scenario we propose, the system is expected to return values for multiple properties at once. Hence, can be considered a generalization of a single-property extraction task as it can be easily formulated as such. Thus, MPE is reverse-compatible with the single-property extraction, and it is still possible to evaluate models trained in the single-property setting.

Many arguments can be considered in favor of framing the problem as MPE. In a typical business scenario, multiple properties are expected to be extracted from a given document. The bulk inference requires a lower computational budget by a factor proportional to the mean number of properties per article, which makes MPE preferable. Moreover, one can expect that systems trained in such a way will manifest emergent properties resulting from the interaction between properties themselves. Consider the set of property-value pairs:

> date of birth: 1915-01-12, date of death: 1979-05-02, place of birth: Saint Petersburg

already predicted by an autoregressive model. It is in principle possible to answer:

> country of citizenship: Russian Empire, country of citizenship: Soviet Union

using the earlier predicted pairs only. This phenomenon emerges if the model (or person) learned the relationships between years, administrative boundaries of the city, and the transformation of the Russian Empire into a communist state that occurred in the meantime. Although no such reasoning is required and the problem can be solved by memorizing related co-occurrence patterns, we intend to achieve the mentioned emergent properties.

Feature	WR	WRR
Base unit	property	article
Examples	18.6M	4.1M
Properties/example	1	4.5
Metric	M-F_1	MMP-F_1
Human-annotated test	−	+
Dataset split	random	controlled
Unseen in evaluation	−	+
Article appears in	few splits	one split

Table 3: Selected differences between WR and WRR. Both metrics are described in Section 6.

4 WikiReading Recycled: Novel Dataset for Multi-Property Extraction

The comparison to existing datasets and shared tasks is briefly presented in Table 1, whereas Table 3 focuses on selected differences between WikiReading Recycled and WikiReading.

4.1 Desiderata

Our set of desiderata is based on the following intentions. We wished to introduce the problem of Multi-Property Extraction to evaluate systems that extract any number of given properties at once from the same source text. Our second objective was to ensure that an article may appear in precisely one data split. The third core intention was to introduce an article-centered data objective instead of a property-centric one. Note that an instance of data should be an article with multiple properties. The fourth objective was to ensure that all properties in the test set can be extracted or inferred. The fifth was to keep the validation and test sets within a reasonable size. Moreover, we aim to provide a test set of the highest quality, lacking noise that could arise from automatic processing. Finally, we intended to benchmark the model generalization abilities – the test set contains properties not seen during training, posing a challenge for current state-of-the-art systems.

4.2 Data Collection and Split

The WikiReading Recycled and WikiReading are based on the same data, yet differ in how they are arranged. Instances from the original WikiReading dataset were merged to produce over 4M samples in the MPE paradigm. Instead of performing a random split, we carefully divide the data assuming that 20% of properties should appear solely in the

Subset	Dev	Test-A	Test-B
rare	4.40	5.12	3.16
unseen	5.53	5.34	2.05
categorical	46.63	44.49	66.51
relational	53.36	55.50	33.49
exact match	20.20	20.16	33.67
long articles	50.39	56.15	30.45

Table 4: An average per-article size of the corresponding subsets as a percent of a total number of properties.

test set (more precisely, not seen before in train and validation sets). Around one thousand articles containing properties not seen in the remaining subsets were drafted to achieve the mentioned objective. Similarly, properties unique for the validation set were introduced to enable approximation of the test set performance without disclosing particular labels. Additionally, test and validation sets share 10% of the properties that do not appear in the train set, increasing the size of these subsets by 2,000 articles each. Another 2,000 articles containing the same properties as the train set were added to each of the validation and test sets. All the remaining articles were used to produce the training set.

To sum up, we achieved a design where as much as 50% of the properties cannot be seen in the training split, while the remaining 50% of the properties can appear in any split. We chose these properties carefully so that the size of the test and validation sets does not exceed 5,000 articles.

4.3 Human Annotation

The quality of test sets plays a pivotal role in reasoning about a system's performance. Therefore, a group of annotators went through the instances of the test set and assessed whether the value either appeared in the article or can be inferred from it. To make further analysis possible, we provide both datasets, before (test-A) and after (test-B) annotation.

The annotation process was non-trivial due to vagueness of the inferability definition, and the scientific character of the considered text. It was required to understand advanced encyclopedic articles e.g., about chemistry, biology, or astronomy, to answer domain-specific properties (scientific classifications or biological taxonomy), which are only possible with deep knowledge about the world and with the ability to learn during the process. Moreover, linguistic skills were required to transliterate and transcribe first and last names. Note that we consider the value which appears in a different writing script as inferable. Due to the stated issues, we decided to rely on highly trained linguists as annotators.

The process was supported by several heuristics. In particular, the approximate string matching was used to highlight fragments of presumably high importance. Nevertheless, it took seven linguists more than 100 hours in total to complete. On average, two minutes and thirty second were required to verify data assigned to one Wikipedia article.

The relevance of annotation mentioned above can be demonstrated by the fact that 28% of the property-value pairs were marked as unanswerable and removed. As it will be shown later, the Mean-Multi-Property-F_1 on a pre-verified test-A was approximately 20 points lower, and 8% of articles were removed entirely from the test-B during the annotation process.

4.4 Diagnostic Subsets

We determined auxiliary validation subsets with specific qualities, not only to help improve data analysis but also to provide additional information at different stages of development of a system. The qualities we measure and the definition is provided below.

Rare, unseen. *Rare* and *unseen* properties were distinguished depending on their frequency. The number of occurrences in the train set was below a threshold of 4000 for each in *rare* and was precisely 0 for the *unseen* category.

Categorical, relational. We denote a property as *categorical* if its value set contains a limited number of values; otherwise, it is *relational*. We apply normalized entropy with a threshold of 0.7 to obtain properties that belong to the *categorical* subset. For instance, the *continent* property occurs 20060 times, but with 13 possible values, its normalized entropy equals 0.43; hence it is marked as *categorical*. This splitting method is not ideal, but we wanted to use the same method as in (Hewlett et al., 2016). For example, if the distribution of continents was uniform, the property would have been classified as relational. However, in practice, it almost never happens.

Exact match. The *exact match* category applies to cases where expected value is mentioned directly in the source text.

Long articles. Instances with articles longer than 695 words (threshold qualifying to the top 15% longest articles in the train set) constitute the *long articles* diagnostic set.

Characteristics of different systems can be compared qualitatively by evaluating on these subsets. For instance, the *long articles* subset is challenging for systems that consume truncated inputs. *Unseen* is precisely constructed to assess systems' ability to extract previously not seen properties. On the other hand, *rare* can be viewed as an approximation of the system's performance on a lower-resource downstream extraction task. The *categorical* subset is useful in assessing approaches featuring a classifier, whereas it is suboptimal to use such systems for *relational* due to richer output space. Similarly, the *exact match* can be approached with sequence tagging solutions. The share of each diagnostic subset is presented in Table 4.

5 Model Architectures

We evaluate different model architectures on the WikiReading Recycled dataset. We re-implemented the previously best performing WikiReading model, finetuned pretrained Transformer models, and applied a dual-source model. Their competitiveness can be demonstrated by the fact that we were able to outperform the previous state-of-the-art on the WikiReading by a far margin.

Basic seq2seq. A straightforward approach to single-property extraction is to use an LSTM sequence-to-sequence model where the input consists of a property name concatenated with the considered input text. To compare with the previous results, we reproduced the basic sequence-to-sequence model proposed by Hewlett et al. (2016).

Vanilla Transformer. A more up-to-date solution is to use the Transformer architecture (Vaswani et al., 2017) instead of an RNN, and a subword tokenization method, such as unigram LM tokenization (Kudo, 2018). We use the term *vanilla* to denote a model trained from scratch.

Vanilla Dual-Source Transformer. The Transformer architecture was extended to support two inputs and successfully applied in Automatic Post-Editing (Junczys-Dowmunt and Grundkiewicz, 2018). We propose to reuse this Dual-Source Transformer architecture in the property extraction tasks.

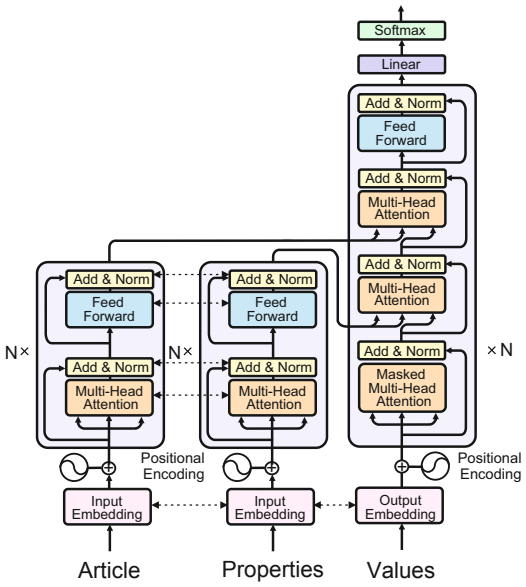

Figure 1: The architecture of Dual-Source Transformer as proposed by Junczys-Dowmunt and Grundkiewicz (2018) for Automatic Post-Editing. In the case of WikiReading Recycled and WikiReading, the encoder transforms an article and the corresponding properties separately.

The architecture consists of two encoders that share parameters and a single decoder. Moreover, the encoders and decoder share embeddings and vocabulary. In our approach, the first encoder is fed with the text of an article, and the second one takes the names of properties (Figure 1). The model is trained to generate a sequence of pairs: (*property*, *value*) separated with a special symbol.

Dual-Source RoBERTa. Recent research shows that pretrained language models can improve performance on downstream tasks (Radford et al., 2018). Therefore, we experimented with the pretrained RoBERTa language model as an encoder. RoBERTa models were developed as a hyper-optimized version of BERT with a byte-level BPE and a considerably larger dictionary (Liu et al., 2019; Devlin et al., 2019). All the model parameters, including the RoBERTa weights, were further optimized on the WikiReading Recycled task.

T5. Recently proposed T5 model (Raffel et al., 2020) is a Transformer model pretrained on a cleaned version of CommonCrawl. T5 is famous for achieving excellent performance on the SuperGLUE benchmark (Wang et al., 2019).

To create a model input, we concatenate a property name and an article. In the case of MPE, we reduce the dataset to the single property setting, as

	Basic seq2seq	Vanilla Transformer	Vanilla Dual-Source	Dual-Source RoBERTa	T5
Numer of inputs	1	1	2	2	1
Pretrained encoder	−	−	−	+	+
Pretrained decoder	−	−	−	−	+
Number of parameters	32M	46M	25M	234M	60M

Table 5: Comparison of evaluated models. The T5 model can be considered as a pretrained equivalent of Vanilla Transformer, and our RoBERTa-based model can be viewed as a partially-pretrained Vanilla Dual-Source Transformer. Basic seq2seq is an RNN counterpart of both T5 and Vanilla Transformer.

used by the T5 model's authors.

6 Evaluation

In this section, we describe the evaluation of previously proposed architectures on both WikiReading and WikiReading Recycled datasets. We would like to highlight that the results are not comparable between the two datasets, as they are based on different train/validation/test splits.

6.1 Metrics

The performance of systems is evaluated using the F1 metric, adapted for the WikiReading Recycled format. For WikiReading, Mean-F_1 follows the originally proposed micro-averaged metric and assesses F1 scores for each property instance, averaged over the whole test set.

Let E denote a set of expected property-value pairs and O model-generated property-value pairs. Assuming $|\cdot|$ stands for set cardinality, precision and recall can be formulated as follows:

$$P(E,O) = \frac{|E \cap O|}{|O|},\ R(E,O) = \frac{|E \cap O|}{|E|}$$

Then F_1 is computed as a harmonic mean:

$$F_1(E,O) = 2 \cdot \frac{P(E,O) \cdot R(E,O)}{P(E,O) + R(E,O)}$$

Given a sequence $\mathcal{E} = \{E_1, E_2, .., E_n\}$ of expected answers for n test instances, and associated sequence of predictions $\mathcal{O} = \{O_1, O_2, .., O_n\}$, we calculate Mean-$F_1$ as:

$$\text{Mean-}F_1(\mathcal{E},\mathcal{O}) = \frac{1}{n} \cdot \sum_{i \in [1,n]} F_1(E_i, O_i)$$

In WikiReading Recycled, we adjust the metric to handle many properties in a single test instance. To do that, the E_i and O_i sets contain values from many properties at once and n is equal to the number of articles. Note that in the case of the M-F_1 properties are considered as instances. We call our article-centric metric Mean-Multi-Property-F_1 or in short MMP-F_1.

6.2 Training Details

Since the basic seq2seq model description missed some essential details, they had to be assumed before model training. For example, we supposed that the model consisted of unidirectional LSTMs and truecasing was applied to the output. The rest of the parameters followed the description provided by the authors.

An extensive hyperparameter search was conducted for both Dual-Source Transformers on the WikiReading Recycled task. In the case of the Dual-Source Transformer evaluated on WikiReading we restricted ourselves to hyperparameters following the default values specified in the Marian NMT Toolkit (Junczys-Dowmunt et al., 2018). The only difference was the reduction of encoder and decoder depths to 4.

For the Vanilla Dual-Source Transformer evaluation, both WikiReading and WikiReading Recycled datasets were processed with a SentencePiece model (Kudo, 2018) trained on a concatenated corpus of inputs and outputs with a vocabulary size of 32,000. Dual-Source RoBERTa model is initialized with $\text{RoBERTa}_{\text{BASE}}$ (consisting of 12 encoder layers and a dictionary of 50,000 subword units).

In the case of the T5 model, we keep hyperparameters as close as possible to those used during pretraining. The training continues with restored AdaFactor parameters. We finetuned the *small* version of the model in a supervised-only manner.

We truncate the input to the first 512 tokens for all our models.

Hyperparameter Optimization. Hyperparameters for WikiReading Recycled were optimized

Model	Mean-F_1
Basic s2s (Hewlett et al., 2016)	70.8
Placeholder s2s (Choi et al., 2017)	75.6
SWEAR (Hewlett et al., 2017)	76.8
Basic s2s (our run)	74.8
Vanilla Transformer	79.3
Vanilla Dual-Source Transformer	**82.4**

Table 6: Results on WikiReading (test set). *Basic s2s* denotes the re-implemented model described in Section 6.2.

using the Tree-structured Parzen Estimator algorithm (Bergstra et al., 2011) with additional heuristics and Gaussian priors resulting from the default settings proposed for this sampler in the Optuna framework (Akiba et al., 2019). An evaluation was performed every 8,000 steps, and the validation-based early stopping was applied when no progress was achieved in 3 consecutive validations. The total number of 250 trials was performed for each architecture. Intermediate results of each trial were monitored and used to ensure only the top 10% trials were allowed to continue. Details of the hyperparameter optimization are presented in Appendix A.

6.3 Results on WikiReading

Although the main focus of our evaluation is the WikiReading Recycled dataset; we additionally evaluate whether the Vanilla Dual-Source Transformer can improve the state-of-the-art on WikiReading.

We reproduced the *Basic seq2seq* model. It achieved a Mean-F_1score of 74.8, which is 4 points higher than reported by Hewlett et al. (2016). The difference may be caused by poor optimization in the original work. Our dual-source solution achieves 82.4 and outperforms the previous state-of-the-art model by 5.6 Mean-F_1 points. To measure the impact of using two encoders instead of one, we evaluated the Vanilla Single-source Transformer, which takes a concatenated pair of article and property as its input. Our dual-source model outperformed its single-source counterpart by 3.1 points. Table 6 presents the final results.

6.4 Results on WikiReading Recycled

The results on WikiReading show that the Dual-Source Transformer is beneficial to the Property Extraction task. On WikiReading Recycled, we supplement the evaluation with pretrained models: Dual-Source RoBERTa and T5.

Table 7 presents Mean-Multi-Property-F_1 scores on the annotated test set (test-B). All the transformer-based models outperform the *Basic seq2seq*. The Dual-Source Transformer achieved 77.5 Mean-Multi-Property-F_1. Its pretrained version, Dual-Source RoBERTa, improves the result by 1.4 points. As the T5 model beats the Vanilla Dual-Source Transformer, we may conclude that even though the WikiReading Recycled dataset is very large, the pretraining is crucial for this MPE task. It is worth remembering that the results on WikiReading and WikiReading Recycled are not comparable due to the dissimilarities in metrics and datasets. We will elaborate on that in section 7.

7 Discussion and Analysis

The final scores of transformer-based models differ slightly on WikiReading Recycled. In order to get more insight, we analyze the models on diagnostic sets described in Section 4.4.

Impact of Property Frequency. We provide two diagnostic sets related to property frequency: *unseen* and *rare*. Both dual-source models failed on the *unseen* subset. These models ignored the *unseen* properties from the input and did not generate any answer. The best result was achieved by the T5 model (10.9 points), albeit it still does not meet expectations.

The results on the *rare* subset show that the pretraining makes a difference if properties are infrequent in the train set (Figure 2).

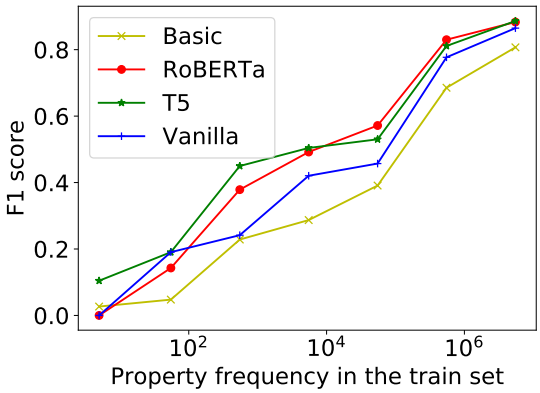

Figure 2: The relation of property frequency and Mean-Multi-Property-F_1. Both RoBERTa and Vanilla refer to Dual-Source Transformers.

Model	unseen	rare	categorical	relational	exact match	long	test-B
Basic seq2seq	2.0	30.2	84.9	50.2	71.1	56.4	75.2
Vanilla Dual-Source	0.0	40.7	83.9	70.8	80.5	63.1	77.5
Dual-Source RoBERTa	0.0	50.7	86.0	76.8	84.3	68.2	80.9
Finetuned T5	10.9	53.8	86.3	73.4	83.4	65.9	80.3

Table 7: Results on WikiReading Recycled human-annotated test set supplemented with scores on diagnostics subsets. All scores are Mean-Multi-Property-F_1.

Impact of Property Type. The extraction of some properties may be treated as a classification task since the set of their valid values is limited. In this case, all models perform similarly and achieve approximately 85 Mean-Multi-Property-F_1. The difficulty of the task increases proportionally to the normalized entropy value, which may lead to the divergence of model performances. This phenomenon is visible in the case of our Basic seq2seq, where the weakness is evident above the 0.5 threshold. The details are presented in Figure 3.

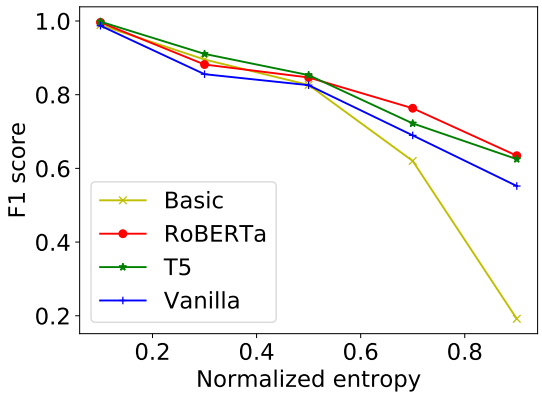

Figure 3: The relation of property normalized entropy and Mean-Multi-Property-F_1. Both RoBERTa and Vanilla refer to Dual-Source Transformers.

Exact Match and Long Articles. The results from the exact match and long articles subsets are correlated with the scores attained on the test-B set; however, the absolute values achieved differ substantially. This is because the long article subset is more challenging, as the chance of an answer appearing in the constant-length prefix decreases with the length of the article. The use of recently introduced models like LongFormer (Beltagy et al., 2020) and BigBird (Zaheer et al., 2020) might decrease the gap in scores between long and average-length articles. On the other hand, system performance should increase when the answer is provided directly in the text, as can be found in the exact match subset.

Difficulty of Test Sets. To compare the difficulty of the WikiReading and WikiReading Recycled test sets, we converted the outputs from the non-annotated WikiReading Recycled test set (test-A) to WikiReading format, and calculated the Mean-F_1. With the Vanilla Dual-Source Transformer, we obtained 54.0 Mean-F_1, 28.4 points less than on WikiReading. This considerable decrease in score shows that the WikiReading Recycled test-A set is more difficult than WikiReading. The reason behind this is that we removed leakage of articles between splits, and we also added more infrequent properties that are harder to answer.

Impact of Human Annotation. The Vanilla Dual-Source Transformer was evaluated on both WikiReading Recycled test sets. It obtained Mean-Multi-Property-F_1 of 62.6 on the non annotated test-A set, while achieving 77.5 on the annotated test-B. This discrepancy suggests that the linguists indeed succeeded to remove non-inferable properties. We anticipate that cleaning the train set in a similar fashion could improve the stability of the training and the overall results.

8 Summary

We introduced WikiReading Recycled—the first Multi-Property Extraction dataset with a human-annotated test set. We provided strong baselines that improved the current state-of-the-art on WikiReading by a large margin. The best-performing architecture was successfully adapted from Automatic Post-Editing systems. We show that using pretrained language models increases the performance on the WikiReading Recycled dataset significantly, despite its large size.

Additionally, we created diagnostic subsets to qualitatively assess model performance. The results on a challenging subset of *unseen* properties reveal that despite high overall scores, the evaluated systems fail to provide satisfactory performance.

Low scores indicate an opportunity to improve, as these properties were verified by annotators and are expected to be answerable. We look forward to seeing models closing this gap and leading to remarkable progress in Machine Reading Comprehension.

The dataset and models, as well as their detailed configurations required for reproducibility, are publicly available.

Acknowledgements

The Smart Growth Operational Programme supported this research under project no. POIR.01.01.01-00-0877/19 (A universal platform for robotic automation of processes requiring text comprehension, with a unique level of implementation and service automation).

References

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, pages 2623–2631, New York, NY, USA. Association for Computing Machinery.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *Computing Research Repository*, arXiv:2004.05150.

James S. Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. 2011. Algorithms for hyper-parameter optimization. In J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 24*, pages 2546–2554. Curran Associates, Inc.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Eunsol Choi, Daniel Hewlett, Jakob Uszkoreit, Illia Polosukhin, Alexandre Lacoste, and Jonathan Berant. 2017. Coarse-to-Fine Question Answering for Long Documents. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 209–220, Vancouver, Canada. Association for Computational Linguistics.

Junyoung Chung, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. In *NIPS 2014 Workshop on Deep Learning*.

Mark Craven and Johan Kumlien. 1999. Constructing biological knowledge bases by extracting information from text sources. In *Proceedings of the Seventh International Conference on Intelligent Systems for Molecular Biology*, pages 77–86. AAAI Press.

Jeffrey Dalton, Chenyan Xiong, and Jamie Callan. 2020. TREC CAsT 2019: The conversational assistance track overview. *Computing Research Repository*, Arxiv:2003.13624.

Hoa Trang Dang, Diane Kelly, and Jimmy J Lin. 2007. Overview of the TREC 2007 question answering track. In *Trec*, volume 7.

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.

Archana Goyal, Vishal Gupta, and Manish Kumar. 2018. Recent named entity recognition and classification techniques: A systematic review. *Computer Science Review*, 29:21–43.

Daniel Hewlett, Llion Jones, Alexandre Lacoste, and Izzeddin Gur. 2017. Accurate supervised and semi-supervised machine reading for long documents. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2011–2020, Copenhagen, Denmark. Association for Computational Linguistics.

Daniel Hewlett, Alexandre Lacoste, Llion Jones, Illia Polosukhin, Andrew Fandrianto, Jay Han, Matthew Kelcey, and David Berthelot. 2016. WikiReading: A novel large-scale language understanding task over Wikipedia. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1535–1545, Berlin, Germany. Association for Computational Linguistics.

Marcin Junczys-Dowmunt and Roman Grundkiewicz. 2018. MS-UEdin submission to the WMT2018 APE shared task: Dual-source transformer for automatic post-editing. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 822–826, Belgium, Brussels. Association for Computational Linguistics.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. Marian: Fast neural machine translation in C++. In *Proceedings*

of ACL 2018, System Demonstrations, pages 116–121, Melbourne, Australia. Association for Computational Linguistics.

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 66–75, Melbourne, Australia. Association for Computational Linguistics.

J. Li, A. Sun, J. Han, and C. Li. 2020. A survey on deep learning for named entity recognition. *IEEE Transactions on Knowledge and Data Engineering*, pages 1–1.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. *Computing Research Repository*, arXiv:1907.11692.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. Technical report, OpenAI.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia. 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.

Tomasz Stanislawek, Anna Wróblewska, Alicja Wójcicka, Daniel Ziembicki, and Przemyslaw Biecek. 2019. Named entity recognition - is there a glass ceiling? In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 624–633, Hong Kong, China. Association for Computational Linguistics.

Raymond Hendy Susanto, Hai Leong Chieu, and Wei Lu. 2016. Learning to capitalize with character-level recurrent neural networks: An empirical study. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2090–2095, Austin, Texas. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019. SuperGLUE: A stickier benchmark for general-purpose language understanding systems. In *Advances in Neural Information Processing Systems 32*, pages 3266–3280. Curran Associates, Inc.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Wei Wang, Bin Bi, Ming Yan, Chen Wu, Jiangnan Xia, Zuyi Bao, Liwei Peng, and Luo Si. 2020. StructBERT: Incorporating language structures into pretraining for deep language understanding. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*.

Yu Wang and Hongxia Jin. 2019. A deep reinforcement learning based multi-step coarse to fine question answering (MSCQA) system. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7224–7232.

Vikas Yadav and Steven Bethard. 2018. A survey on recent advances in named entity recognition from deep learning models. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2145–2158, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

T. Young, D. Hazarika, S. Poria, and E. Cambria. 2018. Recent trends in deep learning based natural language processing [review article]. *IEEE Computational Intelligence Magazine*, 13(3):55–75.

Manzil Zaheer, Guru Guruganesh, Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. 2020. Big Bird: Transformers for longer sequences. *Computing Research Repository*, arXiv:2007.14062.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, ICCV '15, pages 19–27, USA. IEEE Computer Society.

Parameter	Search space	Vanilla Dual-source	RoBERTa
batch size	$2^{\{6,7,8,9\}}$	2^9	2^9
learning rate	1e-5, 5e-5,.., 1e-2	5e-4	5e-5
lr scheduler	inverse sqrt, linear decay	linear	linear
hidden dropout	$0, 0.1$	0	0.1
attention dropout		0	0.1
activation dropout		0	0
weight decay		0	0.1
encoder layers	$1, .., 6$	2	–
decoder layers		2	6
embedding dim*	$2^{\{5,6,..,9\}}$	2^9	–
ffn embedding dim*	$2^{\{6,7,..,11\}}$	2^7	–
attention heads*	$2^{\{2,3,4,5\}}$	2^3	–
activation function*	ReLU, GELU	ReLU	GELU
learned positional emb*	true, false	false	–
share all emb	true, false	false	–

Table 8: Search space considered and hyperparameters determined as optimal when the validation set of WRR is considered. The * symbol denotes tied hyperparameters set to the same values for both encoder and decoder where applicable. The use of pretrained RoBERTa model resulted in the necessity to stick with several architectural choices signalized by – character.

A Hyperparameter Search

Table 8 summarizes search space considered and hyperparameters determined as optimal when the validation set of WRR is considered.

Hyperparameters for WRR were optimized using the Tree-structured Parzen Estimator with additional heuristics and Gaussian priors resulting from the default settings proposed for this sampler in the Optuna framework. An evaluation was performed every 8,000 steps, and the validation-based early stopping was applied when no progress was achieved in three consecutive validations. Intermediate results of each trial (results from every validation) were monitored and used to stop unpromising training earlier.

The trial was pruned in the case its best intermediate value was in the bottom 90 percentiles among trials at the same step (only the top 10% of trials were allowed to continue the training). This process was disabled until five trials finished.

The total number of 250 trials was performed for each architecture.

B Basic seq2seq Replication Details

Since the basic seq2seq model description missed some essential details, they had to be assumed before model training. For example, we supposed that the model consisted of unidirectional LSTMs. It was trained with mean (per word) cross-entropy loss until no progress was observed for 10 consecutive validations occurring every 10,000 updates. Input and output sequences were tokenized and lowercased. Besides, and truecasing was applied to the output. We use syntok[2] tokenizer and a simple RNN-based truecaser proposed by Susanto et al. (2016). During inference, we used a beam size of 8. The rest of the parameters followed the description provided by the authors.

[2] https://github.com/fnl/syntok