

Improving Zero-Shot Event Extraction via Sentence Simplification

Sneha Mehta
Twitter Cortex
snehamehta@twitter.com

Huzefa Rangwala
George Mason University
rangwala@gmu.edu

Naren Ramakrishnan
Virginia Tech
naren@cs.vt.edu

Abstract

The success of sites such as ACLED and Our World in Data have demonstrated the massive utility of extracting events in structured formats from large volumes of textual data in the form of news, social media, blogs and discussion forums. Event extraction can provide a window into ongoing geopolitical crises and yield actionable intelligence. With the proliferation of large pretrained language models, Machine Reading Comprehension (MRC) has emerged as a new paradigm for event extraction in recent times. In this approach, event argument extraction is framed as an extractive question-answering task. One of the key advantages of the MRC-based approach is its ability to perform zero-shot extraction. However, the problem of long-range dependencies, i.e., large lexical distance between trigger and argument words and the difficulty of processing syntactically complex sentences plague MRC-based approaches. In this paper, we present a general approach to improve the performance of MRC-based event extraction by performing unsupervised sentence simplification guided by the MRC model itself. We evaluate our approach on the ICEWS geopolitical event extraction dataset, with specific attention to ‘Actor’ and ‘Target’ argument roles. We show how such context simplification can improve the performance of MRC-based event extraction by more than 5% for actor extraction and more than 10% for target extraction.

1 Introduction

With the proliferation of social media, microblogs and online news, we are able to gain a real-time understanding of events happening around the world. By ingesting large unstructured datasets and converting them into structured formats such as (actor, event, target) tuples we can make rapid progress in systems for event forecasting (Ramakrishnan et al., 2014), real-time event coding (Saraf and Ramakrishnan, 2016) or other applications that can grant

organizations a strategic advantage. Historically, this has been enabled by efforts such as ICEWS¹ & GDELT². These systems rely on event extraction technology to populate their knowledge bases. Fig. 1 gives an example of an event ‘Bring lawsuit against’ from the ICEWS dataset. Extraction involves identifying entities (businessman, employees) corresponding to argument roles ‘Actor’ and ‘Target’. The *event* is triggered by the predicate ‘sued’ in the figure. However, the extraction technology employed by these systems relies on pattern-

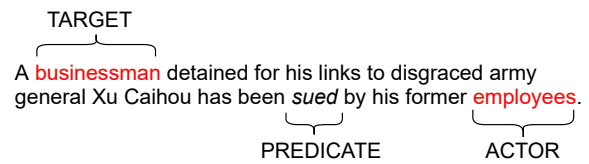


Figure 1: An example of an event of the type ‘Bring lawsuit against’ from the ICEWS dataset.

based approaches that use handcrafted patterns designed to extract entities and events (Bosch et al., 2013). Even though pattern-based methods have high precision, they fail to work on unseen event types and with new event categories. Hence, there is a need to explore extraction methods that can extend beyond a fixed domain. Modern approaches for event extraction (Chen et al., 2015; Nguyen et al., 2016; Wadden et al., 2019) rely on fine-grained annotations and suffer from data scarcity issues and error propagation due to pipeline systems.

With the success of large scale pretrained language models on machine reading comprehension (MRC) tasks (Devlin et al., 2019a; Liu et al., 2019; Huang et al., 2018), a new paradigm for event extraction based on MRC has surfaced (Du and Cardie, 2020; Liu et al., 2020). In this approach, event argument extraction is posed as a span extraction problem from a context conditioned on a question for each

¹<https://dataverse.harvard.edu/dataverse/icews>

²<https://www.gdeltproject.org/>

argument. This approach is promising because it mitigates some of the issues faced by traditional approaches, such as relying on upstream systems to extract entities/triggers and hence sidestepping the error propagation problem in pipeline systems. It also gives rise to the possibility of zero-shot event extraction and hence the ability to extend to new domains which is traditionally hard due to difficulties in collecting high-quality labeled training data. However, MRC models struggle with long-range dependencies and syntactic complexities. For instance, Liu et al. (2020) observe that one typical error from their MRC-based extraction system is related to long-range dependency between an argument and a trigger, accounting for 23.4% errors on the ACE-2005 event dataset (Doddington et al., 2004) (here “long-range” denotes that the distance between a trigger and an argument is greater than or equal to 10 words). Du and Cardie (2020) observe that one of the failure modes of their extraction system is sentences with complex sentence structures containing multiple clauses, each with trigger and arguments. These observations make a promising case for complexity reduction or context simplification for MRC systems.

To mitigate the above problem and to reduce the syntactic complexity we propose an unsupervised approach that is guided by a scoring function that incorporates syntactic fluency, simplicity and the confidence of an MRC model (§ 2) Our key contributions are:

1. The exploration of sentence and context simplification to help mitigate the long-range dependency problem for MRC based zero-shot event extraction (§ 4).
2. Experimental results on the ICEWS political event datasets including a detailed analysis of areas of selective superiority of our approach.
3. Followup analysis to demonstrate how simplification can be controlled based on desired factors such as preference for high performance for a certain argument role.

2 Methodology

Reading comprehension models can be brittle to subtle changes in context. They can be thrown off by syntactic complexity, especially when the questions are not specific and do not include words overlapping with the context. Moreover, long range

dependencies between the trigger/predicate and the argument can also throw the model off. For this purpose, we propose an MRC-guided Unsupervised Sentence Simplification algorithm (RUSS), that iteratively performs deletions and extractions from the context in search for a higher-scoring candidate. The score function incorporates components that ensure sentence fluency, information preservation and the confidence of the target MRC model. Fig. 2 gives an overview of the proposed approach.

Concretely, given that an event has been detected in a sentence, the task is to identify the arguments of the detected event. For instance, in Fig. 1 the task is to identify the arguments ‘Actor’ and ‘Target’ of the event ‘Bring lawsuit against’. The algorithm takes the QA pairs corresponding to each argument role as input. The QA generation procedure for the dataset used in this paper for evaluation is outlined in Appendix B. Table 1 shows the generated QA-pair for the arguments Actor and Target for the event shown in Fig. 1.

Table 1: An example of a generated QA record for an event shown in Fig. 1 from the ICEWS dataset. The highlighted words are answers to the generated questions.

Sentence	A businessman detained for his links to disgraced army general Xu Caihou has been <i>sued</i> by his former employees .
Q-Actor	Who <i>sued</i> someone?
Q-Target	Who was <i>sued</i> by someone?

2.1 Sentence Simplification Algorithm

Given an input sentence s and a list of questions $\{q_1, \dots, q_n\}$ corresponding to different arguments, our algorithm iteratively performs two operations on the sentence – deletion and extraction, in search for a higher-scoring sentence and outputs a candidate simplification c . For generating candidates, the algorithm first obtains the constituency parse tree of the context using a span-based constituency parser (Joshi et al., 2018). It then sequentially performs two operations on the parse tree – deletion and extraction.

Deletion In this operation, the algorithm sequentially drops subtrees from the parse tree corresponding to different phrases. Note that the subtrees with the NP (Noun-Phrase) label are omitted because it is expected that many entities that form event arguments will be noun phrases and deleting them from the sentence would result in significant information

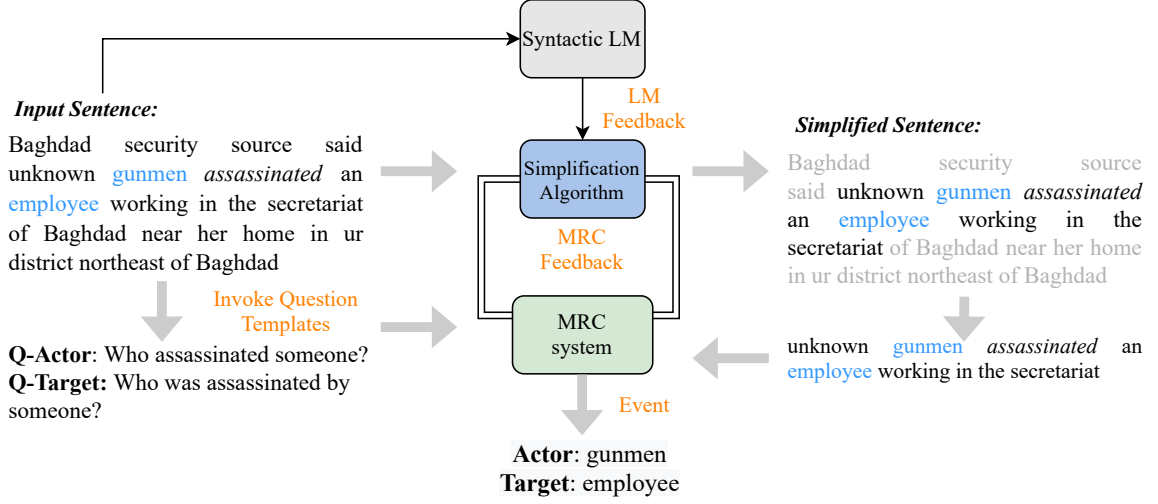


Figure 2: The RUSS sentence simplification approach.

loss.

Extraction This operation simply extracts a phrase, specifically corresponding to the the S and SBAR labels as the candidate sentence. This allows us to select different clauses in a sentence and remove remaining peripheral information.

These operations generate multiple candidates. Candidates with fewer than a threshold of t words are filtered out. We heuristically determine $t = 5$. From the remaining candidates, a highest-scoring candidate is chosen based on the score function described in the next section (§ 2.2). The algorithm terminates if the maximum score assigned to a candidate in the current iteration does not exceed the previous maximum score. The simplification algorithm RUSS is outlined as Algorithm 1 and the candidate generation algorithm is outlined as Algorithm 2.

2.2 Scoring Function

We score a candidate as a product of different scores corresponding to fluency, simplicity and its amenability to the downstream MRC model.

LM Score (ν_{lm}) This score is designed to measure the language fluency and structural simplicity of a candidate sentence. Instead of using LM-perplexity we use the syntactic log-odds ratio (SLOR) (Pauls and Klein, 2012; Carroll et al., 1999) score to measure the fluency. SLOR was also shown to be effective in simplification to enhance text readability (Kann et al., 2018; Kumar et al., 2020). Given a trained language model (LM) and a

Algorithm 1: Sentence Simplification Algorithm – RUSS

Input: $sentence := s, questions = \{q_1, ..q_n\}$
Output: $simplification := c$
Function RUSS(s):
 $maxIter \leftarrow M$
for $iter \in maxIter$ **do**
 $candidates \leftarrow generateCandidates(c)$
 $scores \leftarrow \emptyset$
 $maxScore \leftarrow 0$
 for $cand \in candidates$ **do**
 $scores \leftarrow$
 $scores \cup \nu_{lm}^a * \nu_{entity}^b * \nu_{pred}^c * \prod \nu_{role_i}^{r_i}$
 end
 $currMax \leftarrow max(scores)$
 if $currMax > maxScore$ **then**
 $maxScore \leftarrow currMax$
 $c \leftarrow candidates[argmax(scores)]$
 end
end
return c

sentence s , SLOR is defined as

$$SLOR(s) = \frac{1}{|s|} (\ln(P_{LM}(s)) - \ln(P_U(s))) \quad (1)$$

where P_{LM} is the sentence probability given by the language model, $P_U(s) = \prod_{w \in s} P(w)$ is the product of the unigram probability of a word w in the sentence, and $|s|$ is the sentence length. SLOR essentially penalizes a plain LM’s probability by unigram likelihood and the length. It ensures that the fluency score of a sentence is not penalized by the presence of rare words. A probabilistic language model (LM) is often used as an estimate of sentence fluency. In our work, instead of using a plain LM we use a syntax-aware LM, i.e., in addition to words, we use part-of-speech (POS) and dependency tags as inputs to the LM (Zhao et al.,

Algorithm 2: Candidate Generation Algorithm

Input: $sentence := s$
Output: $candidates$
Function generateCandidates(s):
 $parseTree \leftarrow getParseTree(s)$
 $toRemove \leftarrow \emptyset$
 $extractions \leftarrow \emptyset$
 $candidates \leftarrow \emptyset$
 $phraseTags \leftarrow getValidPhraseTags()$
 for $pos \in parseTree.positions$ **do**
 if $parseTree[pos] \in phraseTags$ **then**
 $toRemove \leftarrow$
 $toRemove \cup parseTree[pos].leaves$
 end
 if $pos.label \in [S, SBAR]$ **then**
 $extractions \leftarrow$
 $extractions \cup parseTree[pos].leaves$
 end
 end
 for $phrase \in toRemove$ **do**
 $candidate \leftarrow s.replace(phrase, \emptyset)$
 if $candidate.length > t$ **then**
 $candidates \leftarrow candidates \cup candidate$
 end
 end
 for $phrase \in extractions$ **do**
 if $phrase.length > t$ **then**
 $candidates \leftarrow candidates \cup candidate$
 end
 end
return $candidates$

2018). For a word w_i , the input to the syntax-aware LM is $[e(w_i); p(w_i); d(w_i)]$, where $e(w_i)$ is the word embedding, $p(w_i)$ is the POS tag embedding, and $d(w_i)$ is the dependency tag embedding. Note that our LM is trained on the original train corpus. Thus, the syntax-aware LM helps to identify candidates that are structurally ungrammatical.

Entity Score (ν_{entity}) Entities help identify the key information of a sentence and therefore are also useful in measuring meaning preservation. The desired argument roles are also entities. Thus, if any entity detected in the original sentence is omitted from a candidate the entity score for that candidate is 0, else it is set to 1.

Predicate Score (ν_{pred}) This score preserves the event predicates in a candidate. It checks if a candidate contains any predicate of interest corresponding to the event detected (Table 5). If it does not then ν_{pred} is set to 0, else it is set to 1.

MRC Score (ν_{rc}) Transformer-based MRC models can be brittle to subtle changes in context. To make the context robust to the MRC model this score allows us to control the complexity of context with respect to the confidence of the MRC model. It is computed separately for each role.

Each argument of an event is a span in the context. $\nu_{rc_{role_i}}^{r_i}$ is the score of the best span in the context for the argument role i , where the score of a candidate span is defined as $ST_x + ET_y$ where $S \in R^H$ is a start vector and $E \in R^H$ is an end vector as defined in Devlin et al. (2019b). T_x and T_y are the final layer representations from the BERT model of the x^{th} and y^{th} tokens in the context. Note that for a valid span, $y > x$. This score is computed separately for each argument role (Actor and Target in Example 1). The importance of the i^{th} role can be controlled by the exponent r_i . The total contribution of each role is computed as the product of score corresponding to each role, given by $\prod \nu_{rc_{role_i}}^{r_i}$.

The final score of a candidate c is computed as follows:

$$\nu(c) = \nu_{lm}(c)^a * \nu_{entity}(c)^b * \nu_{pred}(c)^c * \prod \nu_{rc_{role_i}}^{r_i}(c) \quad (2)$$

Note that b, c can be either 1 or 0 since ν_{entity} and ν_{pred} are binary. In later sections, we evaluate how the simplification can be controlled by varying the constants r_i 's.

3 Datasets and Metrics

We evaluate RUSS on the ICEWS event dataset³ from years 2013 to 2015. In this dataset, event data consists of coded interactions between socio-political actors (i.e., cooperative or hostile actions between individuals, groups, sectors and nation states) mapped to the CAMEO⁴ ontology. These events are in the form of triples consisting of a source actor, an event type (according to the CAMEO taxonomy of events), and a target actor. For evaluation, we aim to extract the Actor and Target roles for event types shown in Table 5. We use the data from years 2013-2015 for training and 2016 for testing/evaluation in section 4.4. See Appendix for distribution of event types.

3.1 Evaluation

We evaluate the performance of an MRC system before and after simplification in the cross-domain zero-shot setting. In this setting, we emulate a no-resource scenario, i.e. using the MRC system out-of-the-box in a target domain. We do not finetune a pretrained MRC model with the generated QA

³<https://dataverse.harvard.edu/dataverse/icews>

⁴<https://parusanalytics.com/eventdata/data.dir/cameo.html>

Table 2: Results of zero-shot event extraction on the ICEWS dataset. ν_{lm} coefficient $a = 1.5$ and ν_{entity} coefficient $b = 1$ for all settings in which simplification is performed. $\Delta +ve$ indicates the % of records for which F1 improves after simplification, $\Delta -ve$ indicates the % of records for which F1 becomes worse after simplification and $\Delta same$ indicates the % of records for which F1 remains unchanged.

	Method	Actor				Target			
		F1	$\Delta +ve$	$\Delta -ve$	$\Delta same$	F1	$\Delta +ve$	$\Delta -ve$	$\Delta same$
1	No simplification	0.412	-	-	-	0.354	-	-	-
2	$c = 0, r_1 = 1, r_2 = 1$	0.431	10.99%	6.54 %	82.45%	0.391	17.35%	7.9%	74.9%
3	$c = 1, r_1 = 0, r_2 = 0$	0.429	10.81%	6.57 %	82.61%	0.390	16.54%	7.53%	75.93%
4	$c = 1, r_1 = 1, r_2 = 1$	0.424	10.5%	6.3 %	83.1%	0.387	16.29%	7.64%	76.05 %
5	$c = 1, r_1 = 3, r_2 = 0$	0.435	9.72%	5.67%	84.6%	0.391	16.89%	7.97%	75.12%
6	$c = 1, r_1 = 0, r_2 = 3$	0.427	10.54%	6.95%	82.5%	0.391	16.12%	7.29%	76.59%

dataset. Rather, the aim is to assess the model performance in a zero-shot setting, without using any training data from the target domain whatsoever. We used the pretrained BERT model finetuned on the SQUAD 2.0 dataset (Rajpurkar et al., 2016) and use the predictor API provided here⁵. We further conduct follow-up analysis to study the controllability of simplification by performing ablation analysis and assessing model performance for different values of score component coefficients. For evaluation, we extracted the best span(s) and computed an exact match F1 score (Seo et al., 2017) matching the span against the ground truth answer.

4 Results & Discussion

The results of zero-shot extraction on the ICEWS dataset are outlined in Table 2. In the baselines used, simplification is performed with score function exponents for ν_{lm} as $a = 1.5$ and ν_{entity} as $b = 1$ held constant while varying c for ν_{pred} , r_1 for ν_{actor} and r_2 for ν_{target} . With no simplification we get F1 scores of 0.412 and 0.354 for actor and target roles respectively. For the most basic setting for simplification with $c = 0$, $r_1 = 1$ and $r_2 = 1$ scores improve by 4.6% for actor prediction to 0.431 and by 10.4% to 0.391 for target prediction respectively which shows that simplifying context can further improve a powerful model like BERT in a cross-domain zero-shot setting. For actor prediction, out of 37,894 records we find that for 10.99% records, F1 score improves after simplification, for 6.54% records F1 decreased after simplification and for the rest the score remained unchanged. For target prediction, for 17.4% records scores improve where as for 7.9% records the scores decreased and for the rest of the records, the scores remained unchanged. After introducing the predi-

cate score ($c = 1$) we see that these improvements drop slightly. This is counter-intuitive, because one would expect model performance to improve when relevant predicates are present in the context. We attribute this behavior to the MRC model leveraging the language priors in the training data to predict the answers. For instance, the model could predict the subject of the predicate as an answer for ‘Who’ type of questions.

Next, we increase the coefficients of Actor and Target roles from 1 to 3. The reason why we choose an odd number for this exponent is because sometimes for bad candidates the RC scores can be negative and since all the scores are combined in a multiplicative way, raising a negative score to an even power would reverse the desired effect. Observing the results in rows 5 & 6 of Table 2 we can see that percentage of sentences with similar scores before and after simplification have increased. We can also observe that percentage of sentences for which scores decrease after simplification have also decreased. We can conclude that by raising the coefficients of role specific scores we can make the simplification models more robust to inaccurate simplifications for those roles. We also observe, when $r_1 = 3$, we get the highest F1 for actor prediction, an improvement of 5.6% over no simplification and for $r_2 = 3$ we can an F1 on-par with the highest obtained in row 2. Our results clearly indicate the benefit of simplification over no simplification and also the gradual improvement in scores when the argument coefficients r_1, r_2 are varied from 0 to 3.

4.1 Long Range Dependencies

Mean length of the original sentences is 32 words where as mean length of the sentences after simplification is 22 words (row 2 setting). This indicates

⁵https://docs.allennlp.org/models/v2.4.0/models/rc/predictors/transformer_qa/

that simplification doesn't make sentences too short as is intuitive because cutting relevant information would harm the performance.

We proceeded to investigate if simplification has addressed the long-range dependency problem. We look at statistics concerning the distance between the predicate and its arguments (Actor and Target) for the setting $c = 0, r_1 = 1, r_2 = 1$, that is, when the predicate score (ν_{pred}) is not taken into account. As Table 2 indicates for 11% of the records performance increases after simplification. We find that for those records the average distance between the predicate and its argument Actor is about 13 words and the average distance between the predicate and target in the simplified context is about 10 words. For the argument Target the average distance between the predicate and target is about 8 words for original and about 6 words for the simplified context.

We see that RUSS cuts about 3 words for Actor prediction and 2 words for Target prediction on average. We conclude that a certain percentage of improvement comes from cutting down the distance between the predicates and arguments hence mitigating the long-range dependency problem.

Moreover, we observe that actor prediction performance is better than that of target prediction. This could be attributed to the fact that for sentences that are in active construction, the subject of the predicate is a candidate for the actor and in active constructions the distance between the predicate and its subject will be small. However, for a complex sentence a clause or a phrase can occur between the subject and the predicate. In this case, the distance between them increases and hence we expect the performance to go down. We quantitatively verify this as follows – the average length between the actor and predicate in the candidate simplification for which the scores improve is about 7 words and for which the scores decreases is about 8 words. Hence, whenever the distance between the predicate and actor is large the performance tends to become worse confirming our hypothesis.

4.2 Qualitative Analysis

Table 3 lists some cases in which simplification helps MRC system perform better. In the first example, the proposed method deleted the word 'personally' from the original sentence (**Sentence**) to obtain the simplified sentence (**Simplified**) as shown

in the Table. The question posed to RC model was "Who is being apologized to by someone" and the ground truth answer is "the opposition". For the original context the model extracts "Nawaz Sharif" as the answer which is the wrong, whereas after removing the adverb "personally", it gets the correct answer. Note, that this decreases the distance between the predicate *apologized* from its argument Nawaz Sharif. In the second example, RC model extracts the closest noun phrase "Xu Caihou" as answer which is incorrect. Simplification deletes the prepositional phrase "to disgraced army general Xu Caihou" aiding the RC model in extracting the correct answer. Note, that in this case it was especially important to delete the above phrase due to the inherent ambiguity of construction. This case also highlights the limitations of the current RC systems as the system was not able to successfully associate employees with businessman and predicted the noun-phrase closest to the predicate *sued*. In the third example, there was segmentation error in the ICEWS dataset and two sentences were strung together as seen in the Table. RUSS successfully deleted the unrelated sentence aiding the RC system in extracting the correct answer.

4.3 Error Analysis

From 6.54% records for which the score decreased after simplification (row 2 of Table 2) for 39.5% records from those the prediction from the original context is a substring of the prediction from the simplified context. This means that for some cases, both the original and the simplified context facilitate the correct answer, rather it is the case that the answer from the simplified context contains extra information for which it is penalized during F1 score computation. For example consider the context "baghdad security source said unknown gunmen assassinated an employee working in the secretariat of baghdad near her home in ur district northeast of baghdad" which after running the simplification algorithm is shortened to "~~in baghdad security source said~~ unknown gunmen assassinated an employee working in the secretariat of baghdad near her home in ur district northeast of baghdad". (The strikethrough text represents the text deleted by the proposed algorithm.) For the question; "Who was assassinated by someone?" when presented with the original context the RC model extracts "an employee" whereas after removing the strikethrough text, RC model extracts "an

Table 3: Qualitative examples of zero-shot performance of RC model before and after simplifying the context using the proposed algorithm. Underlined words are ground truth answers, emphasized words are predicates(triggers) and strikethrough indicates that words were removed by the algorithm.

Question	Who is being apologized to by someone?
Sentence	Islamabad prime minister Nawaz Sharif personally <i>apologized</i> to <u>the opposition</u> today for what he called unfortunate comments made against PPP’s Aitzaz Ahsan
Answer	Nawaz Sharif
Simplified	Islamabad prime minister Nawaz Sharif personally <i>apologized</i> to <u>the opposition</u> today for what he called unfortunate comments made against PPP’s Aitzaz Ahsan
Answer	the opposition
Question	Who is being sued by someone?
Sentence	Scmp a <u>businessman</u> detained for his links to disgraced army general Xu Caihou has been <i>sued</i> by his former employees
Answer	Xu Caihou
Simplified	Scmp a <u>businessman</u> detained for his links to disgraced army general Xu Caihou has been <i>sued</i> by his former employee
Answer	businessman
Question	Who is being accused of something?
Sentence	Thus after having attacked the two elected to his party ump Brice Hortefaux and Claude Goasguen it was accused of pressure and insults. Rachida Dati has <i>accused</i> <u>Claude Goasguen</u> to take to her because she had refused to sleep with him and this during an altercation proved by the Canard Enchan.
Answer	Rachida Dati
Simplified	Thus after having attacked the two elected to his party ump Brice Hortefaux and Claude Goasguen it was accused of pressure and insults. Rachida Dati has <i>accused</i> <u>Claude Goasguen</u> to take to her because she had refused to sleep with him and this during an altercation proved by the Canard Enchan.
Answer	Claude Goasguen

employee working in the secretariat". The ground truth answer for this is "employee". As can be seen both answers are correct but the simplified context is penalized for extra words. Interestingly, such cases make up 48% of records for cases for which performance improves after simplification. This is intuitive, because since context becomes shorter and more precise after simplification and hence one expects RC models to extract more precise answers. The fact that this happens in 39.5% cases in the reverse scenario is surprising.

4.4 In-Domain Training

In sections 4.1- 4.3 we saw how RUSS improved zero-shot event extraction performance in the cross-domain setting. In this section, we consider the scenario when we have labeled in-domain training data available and we wish to investigate if simplification can help improve performance when the MRC system has been finetuned on in-domain data. We use the BERT-base-cased model (Devlin et al., 2019b) as our base model and finetune it on the ICEWS train dataset. We finetune all layers as opposed to just the classification layer as we observe large improvement in the former case as compared to the latter. We use an initial learning rate of 3e-5 and use early stopping with *patience* = 5 to find the best model. For training we use the ICEWS

dataset from years 2013-2015 and the year 2016 for testing. The QA generation procedure is described in section B. There are total 75,788 (37,894×2) examples for training and 5,906 (2,953×2) for test. BERT-RC in Table 4 indicates the performance of the model on the original test set. We use the RUSS algorithm to obtain simplifications of the test set. BERT-RC-Simple indicates the performance of the model on this simplified test set. It can be observed that simplification brings about an improvement(1.4%) even on a model that’s finetuned on in-domain data.

Table 4: Table shows the performance of a BERT-base-uncased model finetuned on in-domain dataset. It can be seen that even after finetuning, RUSS approach improves model performance (BERT-RC-Simple).

Model	F1
BERT-RC	0.776
BERT-RC-Simple	0.787

5 Related Work

Event extraction(EE) has been an active area of research in the past decade. In EE, supervised approaches usually rely on manually labeled training datasets and handcrafted ontologies. Li et al. (2013) utilize the annotated arguments and specific

keyword triggers in text to develop an extractor. Supervised approaches have also been studied using dependency parsing by analyzing the event-argument relations and discourse of event interactions (McClosky et al., 2011). These approaches are usually limited by the availability of the fine-grained labeled data and required elaborately designed features. Recent work formulates event argument extraction as an MRC task. A major challenge with this approach is generating a dataset of QA pairs. Liu et al. (2020) propose a method combining template based and unsupervised machine translation for question generation. Du and Cardie (2020) follow a template approach and show that more natural the constructed questions better the event extraction performance. However, none of these methods directly aim to address the long-range dependency problem using simplification.

Automatic text simplification (ATS) systems aim to transform original texts into their lexically and syntactically simpler variants. The motivation for building the first ATS systems was to improve the performance of machine translation systems and other text processing tasks, e.g. parsing, information retrieval, and summarization (Chandrasekar et al., 1996). In the context of extraction, Zhang et. al. (Zhang et al., 2018) show that pruning dependency trees to remove irrelevant structures can improve relation extraction performance. Efforts have been made to incorporate syntactic dependencies into models in an effort to mitigate this problem 2016; 2018; 2020. Recently, Mehta et al. (2020) have used sentence simplification as a pre-processing step for improving machine translation. Edit-based simplification has been investigated to a great degree to improve the readability of the text (Kumar et al., 2020; Dong et al., 2019; Alva-Manchego et al., 2017). To the best of our knowledge this is the first work that studies sentence simplification for improving MRC-based event extraction.

6 Conclusion & Future Work

In this work, we motivated the need for MRC-based event extraction paradigm especially for zero-shot scenarios (§ 1). Next, we discussed the long-range dependency problem ubiquitously faced by event extraction systems. We proposed a context simplification algorithm to reduce the syntactic complexity of the context aided by MRC-system feedback to address the problem (§ 2). Our results indicate that

simplification can not only aid MRC systems in a zero-shot setting (§ 4.1- 4.3) but also when they’re finetuned on in-domain data (§ 4.4).

Although, the proposed method can be useful to boost extraction performance when offline computations can be afforded, it may be difficult to scale in real-time use cases. Reasons include – 1) RC system inference time while running the algorithm; 2) call latency if using APIs and 3) search algorithms can be computationally expensive. A promising way to improve efficiency would be to generate parallel training data for simplification using the RUSS method offline and guide the MRC model during training using the generated data. This can be done by using attention masks over the deleted words obtained using RUSE. We leave this approach for future work.

Reproducibility: Appendix is provided as supplementary material.

References

- Fernando Alva-Manchego, Joachim Bingel, Gustavo Paetzold, Carolina Scarton, and Lucia Specia. 2017. [Learning how to simplify from explicit labeling of complex-simplified text pairs](#). In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 295–305, Taipei, Taiwan. Asian Federation of Natural Language Processing.
- Elizabeth Boschee, Premkumar Natarajan, and Ralph Weischedel. 2013. *Automatic Extraction of Events from Open Source Text for Predictive Forecasting*, pages 51–67. Springer New York, New York, NY.
- John Carroll, Guido Minnen, Darren Pearce, Yvonne Canning, Siobhan Devlin, and John Tait. 1999. [Simplifying text for language-impaired readers](#). In *Ninth Conference of the European Chapter of the Association for Computational Linguistics*, pages 269–270, Bergen, Norway. Association for Computational Linguistics.
- Raman Chandrasekar, Christine Doran, and Bangalore Srinivas. 1996. Motivations and methods for text simplification. In *Proceedings of the 16th conference on Computational linguistics-Volume 2*, pages 1041–1044. Association for Computational Linguistics.
- Yubo Chen, Liheng Xu, Kang Liu, Daojian Zeng, and Jun Zhao. 2015. Event extraction via dynamic multi-pooling convolutional neural networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics*, volume 1, pages 167–176.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and

- Kristina Toutanova. 2019a. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019b. [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.
- George Doddington, Alexis Mitchell, Mark Przybocki, Lance Ramshaw, Stephanie Strassel, and Ralph Weischedel. 2004. [The automatic content extraction \(ACE\) program – tasks, data, and evaluation](#). In *LREC’04*, Lisbon, Portugal. European Language Resources Association (ELRA).
- Yue Dong, Zichao Li, Mehdi Rezagholizadeh, and Jackie Chi Kit Cheung. 2019. [EditNTS: An neural programmer-interpreter model for sentence simplification through explicit editing](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3393–3402, Florence, Italy. Association for Computational Linguistics.
- Xinya Du and Claire Cardie. 2020. [Event extraction by answering \(almost\) natural questions](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 671–683, Online. Association for Computational Linguistics.
- Lifu Huang, Heng Ji, Kyunghyun Cho, Ido Dagan, Sebastian Riedel, and Clare Voss. 2018. [Zero-shot transfer learning for event extraction](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2160–2170, Melbourne, Australia. Association for Computational Linguistics.
- Vidur Joshi, Matthew Peters, and Mark Hopkins. 2018. [Extending a parser to distant domains using a few dozen partially annotated examples](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1190–1199, Melbourne, Australia. Association for Computational Linguistics.
- Katharina Kann, Sascha Rothe, and Katja Filippova. 2018. [Sentence-level fluency evaluation: References help, but can be spared!](#) In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 313–323, Brussels, Belgium. Association for Computational Linguistics.
- Dhruv Kumar, Lili Mou, Lukasz Golab, and Olga Vechtomova. 2020. Iterative edit-based unsupervised sentence simplification. *arXiv preprint arXiv:2006.09639*.
- Qi Li, Heng Ji, and Liang Huang. 2013. Joint event extraction via structured prediction with global features. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 73–82.
- Jian Liu, Yubo Chen, Kang Liu, Wei Bi, and Xiaojiang Liu. 2020. [Event extraction as machine reading comprehension](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1641–1651, Online. Association for Computational Linguistics.
- Xiao Liu, Zhunchen Luo, and Heyan Huang. 2018. [Jointly multiple events extraction via attention-based graph information aggregation](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1247–1256, Brussels, Belgium. Association for Computational Linguistics.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, et al. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.
- Jie Ma, Shuai Wang, Rishita Anubhai, Miguel Ballesteros, and Yaser Al-Onaizan. 2020. [Resource-enhanced neural model for event argument extraction](#). In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3554–3559, Online. Association for Computational Linguistics.
- David McClosky, Mihai Surdeanu, and Christopher D Manning. 2011. Event extraction as dependency parsing. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, pages 1626–1635.
- Sneha Mehta, Bahareh Azarnoush, Boris Chen, Avneesh Saluja, Vinith Misra, Ballav Bihani, and Ritwik Kumar. 2020. [Simplify-then-translate: Automatic preprocessing for black-box machine translation](#).
- Thien Huu Nguyen, Kyunghyun Cho, and Ralph Grishman. 2016. [Joint event extraction via recurrent neural networks](#). In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 300–309, San Diego, California. Association for Computational Linguistics.
- Adam Pauls and Dan Klein. 2012. [Large-scale syntactic language modeling with treelets](#). In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 959–968, Jeju Island, Korea. 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.
- Naren Ramakrishnan, Patrick Butler, Sathappan Muthiah, Nathan Self, Rupinder Khandpur, Parang Saraf, Wei Wang, Jose Cadena, Anil Vullikanti, Gizem Korkmaz, Chris Kuhlman, Achla Marathe, Liang Zhao, Ting Hua, Feng Chen, Chang Tien Lu, Bert Huang, Aravind Srinivasan, Khoa Trinh, Lise Getoor, Graham Katz, Andy Doyle, Chris Ackermann, Ilya Zavorin, Jim Ford, Kristen Summers, Youssef Fayed, Jaime Arredondo, Dipak Gupta, and David Mares.

2014. [‘beating the news’ with embers: Forecasting civil unrest using open source indicators](#). In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD ’14, page 1799–1808, New York, NY, USA. Association for Computing Machinery.

Parang Saraf and Naren Ramakrishnan. 2016. Embers autogs: Automated coding of civil unrest events. In *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 599–608.

Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2017. Bidirectional attention flow for machine comprehension. *ArXiv*, abs/1611.01603.

Lei Sha, Jing Liu, Chin-Yew Lin, Sujian Li, Baobao Chang, and Zhifang Sui. 2016. [RBPB: Regularization-based pattern balancing method for event extraction](#). In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1224–1234, Berlin, Germany. Association for Computational Linguistics.

David Wadden, Ulme Wennberg, Yi Luan, and Hannaneh Hajishirzi. 2019. [Entity, relation, and event extraction with contextualized span representations](#). 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)*, pages 5784–5789, Hong Kong, China. Association for Computational Linguistics.

Yuhao Zhang, Peng Qi, and Christopher D. Manning. 2018. [Graph convolution over pruned dependency trees improves relation extraction](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2205–2215, Brussels, Belgium. Association for Computational Linguistics.

Yang Zhao, Zhiyuan Luo, and Akiko Aizawa. 2018. A language model based evaluator for sentence compression. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 170–175.

Table 5: Table lists the ICEWS event types used and their corresponding predicates that were identified for generating question templates.

Event Type	Predicates
Abduct, hijack, or take hostage	kidnapped, abducting, abducted, captured
Accuse	blame, blaming, accused, alleged, accusing
Apologize	apologize, apology
Assassinate	carried out assassination of, assassinate
Bring lawsuit against	is suing someone, sued, has sued, filed a suit against
Demonstrate or rally	condemn, protest, demonstrate
Arrest, detain, or charge with legal action	arrested, sentenced, detained, nabbed, captured, arresting, capture, jailed, routinely arrested, prosecuted, convicted
Use conventional military force	killed, shelled, combating, shells, strikes, strike, kill

A Appendix

For training the RUSS algorithm we used the TransformerQA model made available through the allennlp library predictors API ⁶. Running the algorithm takes 5 hours on 1 CPU core and 1 GPU. However when parallelizing the computation across 5 cores that time can be brought down to 1 hour.

B QA Dataset Generation

For the ICEWS dataset, we create one question template per predicate per event type for each slot. For each event type we identified a list of most common predicates (triggers) for that event type since trigger labels are not available in the ICEWS dataset. For example, for ‘Demonstrate or rally’ event type the predicates identified are ‘condemn’, ‘protest’, ‘demonstrate’ and for ‘Accuse’ event type the predicates are ‘blame’, ‘blaming’, ‘accused’, ‘alleged’, ‘accusing’. Table 5 enumerates all of the ICEWS event types used and their identified predicates. For each of the predicates identified for each event type we use one question template for each of the two argument roles Actor and Target. For the Actor role, the template used an active construction ‘Who \$predicate\$ someone?’ and for the same event for the Target role the templated used a passive construction – ‘Who was \$predicate\$ by someone?’. This results in 37,894 records with a sentence and two questions one each for the Actor and Target roles respectively. The list of entities for the ICEWS dataset comes from the ICEWS actors and agents dictionaries⁷.

C Dataset Statistics

Table 6 outlines the distribution of different event types used in the ICEWS dataset used.

⁶https://github.com/allenai/allennlp-models/blob/main/allennlp_models/rc/models/ttransformer_qa.py

⁷<https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/28118>

Table 6: Table shows the distribution of event types in the ICEWS Train and Test datasets used.

Event Type	#Records Train	#Records Test
Abduct, hijack, or take hostage	3473	193
Accuse	8856	651
Apologize	181	11
Arrest, detain, or charge with legal action	9933	782
Assassinate	146	12
Bring lawsuit against	206	18
Demonstrate or rally	2890	175
Use conventional military force	12209	1111