Confli-T5: An AutoPrompt Pipeline for Conflict Related Text Augmentation

Erick Skorupa Parolin*, Yibo Hu*, Latifur Khan*, Patrick T. Brandt†, Javier Osorio‡, Vito D'Orazio†
*Department of Computer Science**, *School of Economic, Political, and Policy Sciences*†
The University of Texas at Dallas, Richardson, Texas
School of Government and Public Policy‡, *University of Arizona*, Tucson, Arizona
{erick.skorupaparolin, yibo.hu, lkhan, pbrandt, dorazio}@utdallas.edu, josorio1@arizona.edu

2022 IEEE International Conference on Big Data (Big Data) | 978-1-6654-8045-1/22/$31.00 ©2022 IEEE | DOI: 10.1109/BigData55660.2022.10020509

***Abstract*—Recent advances in natural language processing (NLP) and Big Data technologies have been crucial for scientists to analyze political unrest and violence, prevent harm, and promote global conflict management. Government agencies and public security organizations have invested heavily in deep learning-based applications to study global conflicts and political violence. However, such applications involving text classification, information extraction, and other NLP-related tasks require extensive human efforts in annotating/labeling texts. While limited labeled data may drastically hurt the models' performance (overfitting), large demands on annotation tasks may turn real-world applications impracticable. To address this problem, we propose Confli-T5, a prompt-based method that leverages the domain knowledge from existing political science ontology to generate synthetic but realistic labeled text samples in the conflict and mediation domain. Our model allows generating textual data from the ground up and employs our novel *Double Random Sampling* mechanism to improve the quality (coherency and consistency) of the generated samples. We conduct experiments over six standard datasets relevant to political science studies to show the superiority of Confli-T5. Our codes are publicly available [1].**

***Index Terms*—text augmentation, generation, classification, natural language processing, conflict, coding event data, CAMEO**

I. Introduction

Political scientists and government agencies in the security sector have invested large resources on analyzing conflicts and political violence across the globe. Extracting information and discovering knowledge from extensive unstructured data (news articles) are crucial to monitoring, understanding, and predicting the dynamics of social unrest, political violence, and armed conflict worldwide.

During the past two decades, political scientists and computational linguistics have explored two main directions to extract structured event data from news articles. First, *pattern-matching* based approaches such as PETRARCH family [1]–[3] have been used to capture conflict interactions from text and convert them to the form of a who-did-what-to-whom template. These approaches rely on external repositories to identify the presence of certain lexico-syntactic patterns in natural language sentences. In the second (and more promising) direction, *statistical language modeling* approaches exploring natural language processing (NLP) techniques have been designed to address information extraction (IE), text classification and other tasks in political science and conflict domains.

Recent advances in deep learning and computational linguistics have pushed political science scholars to focus their efforts in the second direction. Previous efforts employing transformer-based pre-trained language models (PLMs) [4]–[8] have shown successful results in several political science subareas, such as organized crime [9], protests [10], and general conflict and mediation topics [11]–[13].

However, most political and social science applications involving text classification, information extraction, or other NLP-related tasks require extensive human efforts in annotating texts. Limited labeled data over-fits supervised deep learning models, drastically hurting their performance. On the other hand, the need for large amounts of resources (time and money) and expertise to obtain enough labeled data may preclude the application of such powerful models in real-world cases.

To address this problem, we propose Confli-T5, a pipeline model for generating synthetic text samples in the conflict and mediation domain. Confli-T5 is a prompt-based model that explores the knowledge resting in CAMEO (the most prominent ontology and industry standard on political science) through the large-scale language model T5 [7] to generate synthetic labeled data for text classification. Our method differs from previous augmentation models by dispensing human inputs on prompt engineering and maintaining the consistency between augmented text and their labels. We conduct extensive experiments on six standard datasets relevant to conflict research to demonstrate the superiority of our method.

This paper makes multiple contributions, bridging deep learning for big data and geopolitics to support the advances in conflict analysis. First, to the best of our knowledge, we are the first to propose a prompt-based model that transfers learning from a complex ontology (and its knowledge bases) for text augmentation. Second, our model generates labeled textual samples without requiring pre-existing labeled data (as the other baseline models do). Third, we introduce an innovative approach called *double random sampling* to improve the coherence and consistency of the generated synthetic text. Finally, we conduct extensive experiments applied to political sciences to compare with existing text augmentation methods.

[1] https://github.com/erickparolin/Confli-T5

II. Preliminaries

A. Related Work

Prompt-based Learning. Prompt engineering in NLP involves embedding the task description as part of the input sequence. For example, references [11], [14]–[18] add task-specific prefixes in the dataset to train a language model for zero/few-shot or text generation purposes. In our application, we use prompt engineering to design a template for input sequences that favors data augmentation for text classification. Our method differs from other prompt-based generation methods by dispensing human inputs to design the prompts. Confli-T5 automatically constructs prompts by resorting to existing ontology, making prompt engineering more simple and efficient.

Text Augmentation. Generating synthetic text data alleviates extensive costs (time, money, and expertise) associated with annotating texts. However, text augmentation is not a simple task once it involves attending complex syntactic and semantic structures. Previous works have explored text augmentation approaches based on synonym replacement [19], [20] and paraphrasing techniques based on back-translation [21]–[23]. Other works explored large-scale language models by prepending the existing class labels to input sequences [24], perturbing latent spaces [25]–[28], or employing masked language models as denoising autoencoders [29]. Recent mix-up approaches [17], [30]–[32] mix pre-existing samples or interpolate them in hidden spaces to produce realistic texts.

Our model differs from the other augmentation methods in two crucial aspects. First, it allows labeled text generation dispensing pre-existing annotated data by exploring an existing ontology. Second, Confli-T5 maintains the consistency between the generated texts and the associated labels (through our *double random sampling* method). In this way, we mitigate noisy data points and improve text classification.

Coding Political Event Data. Coding events consists of extracting structured data from news articles, usually in the who-did-what-to-whom format. Previous works include pattern-matching approaches [1]–[3], classical machine learning [33]–[36], transformer-based networks [12], [13], and other deep learning methods [37]–[39]. Next, we briefly describe CAMEO, the industry-standard schema for event extraction in political sciences.

B. CAMEO: Conflict and Mediation Observations

CAMEO is a dominant ontology for political event data that incorporates data repositories for **action-pattern** dictionaries (≈ 14K entries) and **actor** dictionaries (≈ 67K entries).

The action-pattern repository contains verbal patterns (resembling regular expressions) associated with political interaction categories (CAMEO codes). Despite the high granularity of event types covered by CAMEO (more than 200 codes), conflict scholars traditionally use a higher level of categories, grouping the original types into 20 (rootcodes) or 5 classes (pentacodes), as detailed in CAMEO **codebook**[2].

Take the following action-pattern as an example:

```
$ * ROCKET_ATTACK + [194]     # LAUNCH
```

This action-pattern is based on the verb *launch* and indicates that occurrences in news articles matching this pattern should be categorized with CAMEO code 194, which corresponds to rootcode 19 and pentacode 4. In this example, symbols **$** and **+** refer to *source* (subject) and *target* (object) of the action, respectively. The symbol * indicates where the verb must occur (in any tense) in the pattern. Additional words surrounding the tokens in the pattern will not change the action code 194, unless they occur between the tokens linked by the symbol _ (e.g., "... rocket **and** attack ...").

The actor repositories store information about political entities and their corresponding roles. Entities can be politicians (persons); parties, gangs, associations or organizations (group); and even political agents representing countries or cities (place). The following is an entry from actor repository:

```
JUHA_KORKEAOJA [FINGOVAGR 030501-070430]
```

This entry stores information about a politician called Juha Korkeaoja, who was Minister of Agriculture (code GOVAGR) of Finland (code FIN) between 2003 and 2007.

In summary, CAMEO is a static ontology where the knowledge rests. As aforementioned in the previous subsection, pattern-matching systems (e.g., PETRARCH) rely on CAMEO to syntactically explore input sentences, looking for matches of action-patterns and actors.

III. Method

Following, we describe the components of Confli-T5. As depicted in Fig. 1, it first leverages CAMEO to automatically produce prompts based on the knowledge resting in this ontology. Next, the natural language generation (NLG) model T5 [7] generates synthetic labeled texts. Then, BART [6] works as a natural language inference (NLI) parser to improve the quality of the generated data. Finally, the generated data serves as augmented data to train a supervised model for a downstream task. This paper focuses on text generation for classification purposes, leaving the analysis on other NLP tasks as future work.

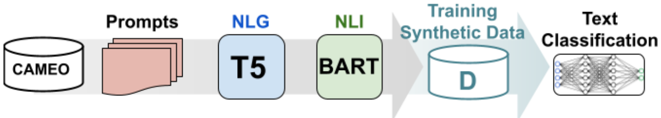

Fig. 1: Diagram of text augmentation with Confli-T5.

A. CAMEO-based Prompts

Before describing the procedure to construct the prompts, we formalize the following rules previously discussed in

[2] https://parusanalytics.com/eventdata/data.dir/cameo.html

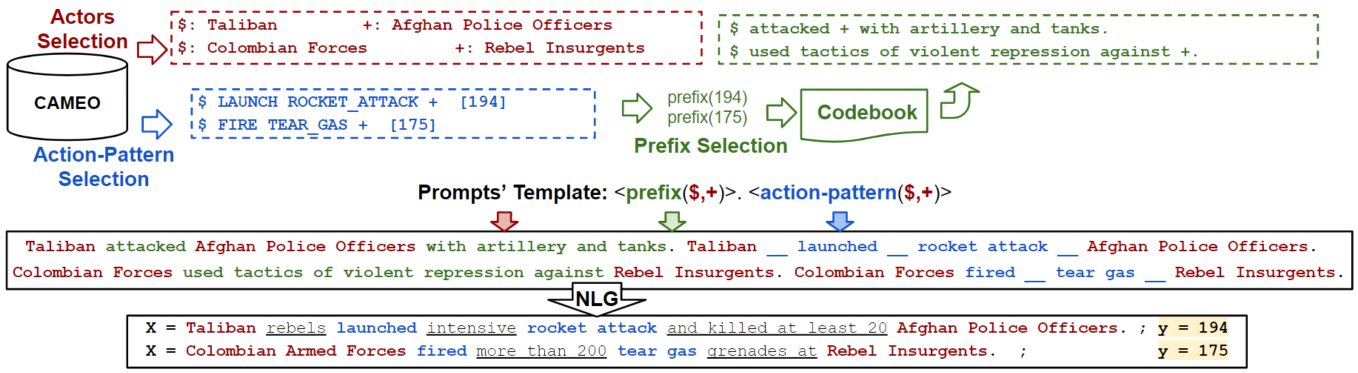
Actors Selection
$: Taliban +: Afghan Police Officers
$: Colombian Forces +: Rebel Insurgents
$ attacked + with artillery and tanks.
$ used tactics of violent repression against +.
CAMEO
$ LAUNCH ROCKET_ATTACK + [194]
$ FIRE TEAR_GAS + [175]
prefix(194)
prefix(175)
Codebook
Prefix Selection
Action-Pattern Selection
Prompts' Template: <prefix($,+)>. <action-pattern($,+)>
Taliban attacked Afghan Police Officers with artillery and tanks. Taliban __ launched __ rocket attack __ Afghan Police Officers.
Colombian Forces used tactics of violent repression against Rebel Insurgents. Colombian Forces fired __ tear gas __ Rebel Insurgents.
NLG
X = Taliban rebels launched intensive rocket attack and killed at least 20 Afghan Police Officers. ; y = 194
X = Colombian Armed Forces fired more than 200 tear gas grenades at Rebel Insurgents. ; y = 175

Fig. 2: Actors and action-patterns are randomly selected from CAMEO ontology. The prompt prefixes are selected based on the action-codes. Actions, actors and prefixes will then fill the prompts' template to construct the prompts (including blanks). The prompts will feed the NLG model, which in turn fills the blanks to generate synthetic labeled samples for text classification.

Subsection II-B: every political actor ρ stored in CAMEO is associated with an actor code $code_p(\rho)$; as well as an action-pattern ν is mapped to a CAMEO code, defined as $code_v(\nu)$.

Fig. 2 illustrates the steps to construct the prompts and the corresponding template with two real examples. The prompts ' template is composed of three parts: action-pattern ν, actors (source ρ_{src} represented by \$ and target ρ_{tgt} represented by +), and prefix. The prefix consists of a brief description of the action code $code_v(\nu)$ extracted from the CAMEO codebook.

Our procedure depicted in Fig. 2 first randomly selects an action-pattern ν from the action-pattern dictionary. Then, it selects the source ρ_{src} (subject of the action ν) and target ρ_{tgt} (object of ν) from actors dictionary. Actor ρ_{src} is randomly selected from set $\{\rho \mid code_p(\rho) = code_{src}\}$. The code $code_{src}$ is selected according to the conditional probability distribution $P(src = code_{src} \mid code_v(\nu))$, which denotes the probability that any political actor associated with code $code_{src}$ appears as the source of any action ν with code $code_v(\nu)$. The actor ρ_{tgt} is selected similarly with $P(tgt = code_{src} \mid code_v(\nu))$ instead. These conditional probabilities were pre-computed based on statistics observed on a dataset available from a previous study [40] (1,920,174 real-world sentences from more than 400 news agencies). After preliminary experiments, we concluded that using these pre-computed distributions produces better results than simply randomly selecting political actors.

Next, the prompt's *prefix* is selected from a dictionary structure $prefix(\cdot)$, which maps an action code $code_v(\nu)$ to the description for this action. As illustrated in Fig. 2, the action-pattern ν="LAUNCH ROCKET_ATTACK +" with $code_v(\nu)$=194 will return the prefix $prefix(194)$ = "*\$ attacked + with artillery and tanks*". Based on our empirical analysis, introducing the action descriptions as prefixes in prompts improves the quality of the text generated.

Lastly, we put the components above together to form the final prompt. As depicted in Fig. 2, prefix and action-pattern are appended and filled up with the selected actors (replacing \$ and + symbols). Blank tokens "__" are added among the action-pattern words to indicate where the NLG model will fill up. The CAMEO codes $code_v(\nu)$ associated with action-patterns ν will later indicate the prompts' labels using ν.

B. Double Random Sampling Strategy

As depicted in Fig. 1, T5 is employed as the NLG model for text infilling on CAMEO-based prompts (as exemplified in Fig. 2). Following, BART will work as an NLI parser to filter out incoherent and inconsistent text generated samples.

Conditional Generation. Technically, auto-regressive language generation models (such as T5) work with the assumption that the probability of a word sequence can be decomposed into the product of conditional next word probabilities:

$$P(w_{1:T} \mid W_0) = \prod_{t=1}^{T} P(w_t \mid w_{1:t-1}, W_0) \tag{1}$$

where W_0 is the initial context and w_t is the word or token to be generated at a given step t in the sequence. For a given vocabulary V, the probability of a word $v_l \in V$ occur in the position w_t of the sequence is:

$$P(w_t = v_l \mid w_{1:t-1}, W_0) = \frac{\exp\left(z_l / temp_1\right)}{\sum_j^V \exp(z_j / temp_1)} \tag{2}$$

where $z_{1:|V|}$ are the logits from language model's output layer and $temp_1$ is the temperature used to re-estimate the softmax above.

In our implementation, we use *nucleus sampling* [41] as a decoding mechanism for text generation with T5. Instead of picking the next token w_t to maximize the probability expressed in Eq. 2, nucleus sampling randomly selects w_t considering the shape of the probability distribution. We select the highest probability tokens whose cumulative probability mass exceeds the threshold p and adjust the original probability distributions for this small subset of vocabulary. From $P(w_t = v \mid w_{1:t-1}, W_0)$, the *top-p* vocabulary $V^{'} \subset V$ is defined as the smallest set such that

$$\sum_{v \in V'} P(w_t = v \mid w_{1:t-1}, W_0) \geq p \tag{3}$$

The original distribution from Eq.2 is re-scaled as follows:

$$P(w_t = v|w_{1:t-1}) = \begin{cases} P(w_t = v|w_{1:t-1})/p^{'}, & \text{if } v \in V^{'} \quad (4) \\ 0, & \text{otherwise} \quad (5) \end{cases}$$

where $p^{'} = \sum_{v \in V^{'}} P(w_t = v \mid w_{1:t-1}, W_0)$.

Nucleus sampling introduces a certain level of randomness in the generated text, making it closer to human-written. The temperature sampling in the softmax equation (Eq. 2) will produce more coherent synthetic samples. It makes the distribution less random and skews towards high-probability events, thus improving the decoding process. Therefore, we use T5 for condition generation with nucleus sampling to fill up the blanks in CAMEO-based prompts (see bottom of Fig. 2) and generate the *full synthetic corpus* $\tilde{D}$.

Natural Language Inference. NLI is a standard NLP task that determines whether a *hypothesis* is true (entailment), false (contradiction), or undetermined (neutral) given a *premise*. Both text sequences for a premise and a hypothesis are given as input to the model. Confli-T5 implements the transformer-based BART for NLI as a zero-shot mechanism to verify whether the generated texts are consistent with the labels (CAMEO codes) assigned to each prompt. For a generated text sample, we take the excerpt corresponding to the prefix (action description from codebook) as hypotheses and the text generated from the action-pattern as a premise. Given the (premise, hypothesis) pair as input, we use BART's entailment score to identify incoherent or inconsistent generated samples.

Table I shows real examples of generated text samples from $\tilde{D}$, followed by their prefixes, original action-patterns, CAMEO codes, and entailment (NLI) scores. In examples corresponding to IDs 1 and 2, the generated texts are consistent with their corresponding prefixes, with the high entailment scores reinforcing such consistency. On the other hand, Ex.IDs 3 and 4 show low entailment scores, indicating either lack of consistency between generated text and prefixes (Ex.ID 4) or a lack of coherence in the generated text (Ex.ID 3).

However, most of the generated samples with the highest scores are short sentences that barely reproduce the CAMEO action-patterns by simply filling them with prepositions and articles (Ex.IDs 1 and 2). While searching examples with slightly lower entailment scores (Ex.IDs 5 and 6), we observed that these generated samples add more tokens over the original patterns while keeping consistency and coherency.

It seems beneficial to eliminate the samples with low NLI scores to avoid noisy examples in the synthetic training data. Besides, searching for distinct and more naturally generated samples will increase the quality and diversity of the training set. Based on these observations, we design an extra layer of random sampling called **top-q sampling** (inspired by top-K [42], [43]) to select the generated sentences from $\tilde{D}$.

Top-q sampling first filters the subset of sentences $Q \subset \tilde{D}$ such that the entailment score is higher than a threshold q. From Q, it constructs the **synthetic training data** $D \subset Q$ by randomly selecting $|D|$ sentences according to probability distribution proportional to the NLI scores and the topics we want to train the supervised model. Thus, the probability of selecting a synthetic sentence $d \in Q\tau$ is

$$P(d) = \frac{\exp\left(nli(d)/temp_2\right)}{\sum_{e}^{Q\tau} \exp(nli(e)/temp_2)} \qquad (6)$$

where $nli(d)$ is the NLI score for d, $Q\tau \subset Q$ is composed only by synthetic samples associated to a topic τ (a CAMEO code), and $temp_2$ is a temperature (as in Eq. 2).

Top-q sampling allows controlling consistency between generated texts and labels (through NLI) while keeping text fluency and diversity provided by nucleus sampling. The usage of prefixes in the prompts is beneficial not only for providing a context (W_0 in Eq. 1) but also for controlling label consistency through top-q. We call the two-layer of random sampling (nucleus and top-q sampling) as **Double Random Sampling**.

C. *Training Synthetic Data*

We close this section by putting together all the steps previously discussed in Algorithm 1. Confli-T5 Procedure receives as input the thresholds p and q (see III-B), temperatures $temp_1$ and $temp_2$, the desired output data size N=$|D|$, an optional pre-existing labeled data Λ, and two dictionaries $CAMEO2labels$ and $CAMEO2distr$.

Algorithm 1: Confli-T5 Procedure

```
input : dictionaries CAMEO2labels and CAMEO2distr,
        thresholds $p$ and $q$, temperatures $temp_1$ and $temp_2$,
        output size N, labeled data $\Lambda$ (default None)
output: training synthetic data $D$
1  explored_codes ← CAMEO2labels.keys()
2  prompts ← get_prompts(CAMEO, explored_codes)
3  $\tilde{D}$ ← T5_generation(prompts, explored_codes, $p$, $temp_1$)
4  foreach $d$ in $\tilde{D}$ do  $d$.nli ← BART_nli(d.text, d.prefix)
5  if $\Lambda$ is not $None$ then  $D \leftarrow \Lambda$
6  else  $D \leftarrow \{\emptyset\}$
7  foreach (code $\tau$, probability $P\tau$) in CAMEO2distr.items() do
8      $y$ ← CAMEO2labels[$\tau$]
9      size ← N * P$\tau$
10     $Q\tau \leftarrow topQFilter\,(\tilde{D},\ \tau,\ q)$
11     $D\tau \leftarrow topQSampling\,(Q\tau,\ size,\ temp_2)$
12     foreach $d$ in $D\tau$ do  $D$.append( ($d$.text, $y$) )
13 return $D$
```

We can add smaller portions of labeled data or additional data outside the conflict domain (e.g., sports, technology, or religion) in D through parameter Λ.

The dictionary CAMEO2labels maps the chosen CAMEO codes to the final desired labels, while CAMEO2distr maps these codes to the desired distributions in the final data D. Line 2 creates the prompts (see III-A), while lines 3 and 4 generate the synthetic samples through T5 and computes NLI score through BART, respectively. Finally, the training data D is constructed in Lines 7 to 12 by top-q searching on D (see III-B) and mapping the pre-selected CAMEO codes to desired

TABLE I: Examples of text samples generated using Confli-T5 and their corresponding prefixes (with sources in red and targets in blue), CAMEO action-patterns (with main verbs in bold), CAMEO codes and entailment (NLI) scores.

Ex. ID	Generated Text (Premise)	Prefix (Hypothesis)	CAMEO Pattern	CAMEO Code	NLI Score
1	Italy lifted ban on trade with Cuba in 2009.	Italy eased economic sanctions on Cuba.	$ **LIFT** BAN ON TRADE WITH +	085	0.9971
2	Prime Minister of the United Kingdom canceled a peace talk with Afghanistan on July 9.	Prime Minister of the United Kingdom halted negotiations with Afghanistan.	$ **CANCEL** PEACE TALK +	164	0.9967
3	Armenian War Vessel died from injuries and damage caused to their crew by U.S. Reaper Drone in December 2017.	U.S. Reaper Drone attacked Armenian War Vessel through conventional military force.	+ **DIE** FROM INJURIES CAUSED BY $	190	0.0002
4	K. Annan decided not to open a formal investigation of the Iraqi ministries.	K. Annan investigated Iraqi ministries.	$ **DECIDE** TO_OPEN INVESTIGATION +	090	0.0063
5	United Nations Commission for Human Rights voiced concern over the Iran refusal to cooperate with Syria.	United Nations Commission for Human Rights disapproved Iran, raising many objections.	$ **VOICE** CONCERN OVER + REFUSAL	110	0.9784
6	Malaysian Minister of Domestic Trade voted in favour of the proposal to strengthen its sanctions against Libya.	Malaysian Minister of Domestic Trade imposed sanctions on Libya.	$ **VOTE** STRENGTHEN SANCTIONS +	163	0.9603

training labels. Text x appended to D in line 12 is composed of generated texts only, discarding prefixes and NLI scores.

IV. Experiments and Results

A. Setup

We ran 10 rounds of the training process for each model with one Quadro RTX 8000 GPU. Then we reported the averaged results observed on the testing set. In each round, we generated different train/validation splits (85%/15% over training data). We randomly initialized the models based on the seed assigned for that round. We trained the models over 20 epochs and selected the best model for each round based on the validation f1-scores. We used the same random seeds for all evaluated models and set the following Confli-T5 hyper-parameters: p=0.9, q=0.975, $temp_1$=0.95 and $temp_2$=0.90. For all the experiments, we utilized the same full synthetic corpus D of size $|D|$=$408,000$ and explored it using top-q search with different topics (CAMEO codes).

As pre-trained language models, we used *t5-large* for T5 and *bart-large-mnli* for zero-shot BART. As transformer-based networks for training the models with synthetic data, we used *bert-base-uncased*.

For a more comprehensive evaluation, we selected three augmentation methods using completely different approaches as baselines. EDA [19] applies simple operations such as synonym replacement, random insertion, swap, and deletion to augment texts. TMix [32] creates augmented training samples by interpolating text hidden space in BERT models. Finally, GPT3Mix [17] (G3M in experiments) is a prompt-based generation method that uses pseudo-labeling to generate text samples with their soft labels. We used the hyperparameters reported by the authors. The data splits, the number of seeds, and reporting approach were the same for all the models evaluated in this section.

B. Datasets

We evaluated the models' performance over six standard datasets used in political and social science studies. As described next, we slightly pre-processed some of the following datasets to utilize them for text classification.

Conflict and Mediation Observations (CAMEO) [13] is a sentence-level dataset following the standard event extraction schema in political science (see II-B). The data points were annotated with the actions (pentacodes) occurring in the sentences. We removed the records associated with pentacode 0 ("Make a Statement") to concentrate our analysis on conflict and mediation topics.

Automatic Content Extraction 2005 (ACE05) is a widely used event-extraction dataset. It annotates 33 event types, including conflict-related subjects (e.g., Attack and Demonstrate labels), which correspond to approximately 30% of the total annotated events. Political and social scientists are interested in extracting conflict-related events from large corpora. Thus, we converted ACE05 to evaluate Confli-T5's data augmentation performance to classify whether the sentence contains conflict-related events.

Massive Event Detection (MAVEN) [44] annotates 168 event types, including military, civil, and terrorist-related conflicts. We utilized the topic labels of documents to split the original document-level data into three conflict categories for text classification, as described in Table II.

WikiEvents (Wiki) [45] is a document-level event extraction dataset containing 50 event types, including conflict-related categories such as violent attack and demonstration. We collected the sentences with conflict-related events with flag 1 and the remaining sentences as 0 (see Table II), similarly as we did for ACE05.

Global Contention Politics (GLOCON) [46] is a sentence-level corpus containing records of real-world protest events reported in distinct countries (e.g., India, China, South Africa, and Argentina). We utilized GLOCON data following the same format used in previous work [12].

India Police Events (IndPol) [47] contains news sentences (in English) from Times of India articles reporting police activity events during a period of widespread Hindu-Muslim violence in Gujarat (March 2002). The sentences were annotated in a multi-label fashion considering four categories of police activity: kill, arrest, fail to act, and force. We removed the data points either containing no police activity events or more than one event.

Table II summarizes the details regarding the organization and pre-processing of these datasets. Information under *Label Mappings* shows which *Rootcodes* we used to synthesize texts associated with *Original* labels in the datasets. Specifically for IndPol data, we used CAMEO codes (in parenthesis) instead of rootcodes level. The last column *Label* denotes the final labels we used in the synthetic training data D. In practice, columns *Rootcodes* and *Label* show the information stored in structure CAMEO2labels in Algorithm 1. In our experiments, we make the distributions in CAMEO2distr follow the same distribution as in the original data.

TABLE II: Datasets description: sizes and mapping from original to rootcodes (or CAMEO codes).

Dataset	**Train/Test**	**Label Mappings**		
		Original	**Rootcodes**	**Label**
CAMEO	1,799/395	Verb. Coop.	3-5	0
		Mat. Coop.	6-8	1
		Verb. Confl.	9-13, 16	2
		Mat. Confl.	14, 15, 17-20	3
ACE05	3,056/766	Attack, Demonstrate	14 and 19	1
		Others	1, 2, 3, 4, 7, 8	0
MAVEN	2,895/725	Mil.Confl./Att./Oper.	15 and 19	1
		Civ.Attack, Civ.Conflict, Terrorist Attack	14	2
		Others	4 and 5	0
Wiki	1,582/396	Conflict	14, 18 and 19	1
		Others	3, 4 and 7	0
GLOCON	1,548/388	Protest	14	1
		No Protest	1-8	0
IndPol	555/140	Kill	(1823,185,186,202)	0
		Arrest	(173)	1
		Fail to Act	5 and 12	2
		Force	(170-175,190-193)	3

C. *Data Augmentation Experiments*

Traditionally, text augmentation methods require a pre-existing portion of annotated text to augment it. So, we randomly sampled the existing training data into smaller portions (e.g., 1%, 5%, 10%, 25%, 50% of the original size) and assumed that these samples were the pre-existing annotated data available. Then, we applied the augmentation methods to synthesize data of different sizes, increasing the pre-existing sample by an *augmentation factor* (e.g., 1× or 2× of the sample size). Finally, we trained the downstream BERT classifiers using the pre-existing plus the synthetic samples as training data. We measured the classification performance using the original test sets. We could add pre-existing data and control the augmentation factor in Confli-T5 through the inputs Λ and N in Algorithm 1, respectively.

Table III shows the f1-scores observed on downstream classification over the six datasets (see IV-B), considering 20 possible scenarios (5 sample sizes × 4 augmentation factors). The values under the column 0× show the f1-scores observed when no augmentation is done (training on the samples only). Furthermore, the lines 0% (of sample) show the performance observed while training the models with synthetic data only with augmentation factors applied over the original data set (instead of the sample sizes). Since the baseline models cannot augment without pre-existing annotated data, then no values are input for them on these lines. Bold values indicate the best f1-score for each sample size and augmentation factor, while underlined values indicate the best performance on classification for the evaluated sample sizes of a dataset. The last line averages the f1-scores measured on all datasets for each augmentation method. Following are the findings from the results in Table III.

Confli-T5 outperforms text augmentation baselines in most cases by a large margin. Our model produces better results in most of the 20 scenarios on all the evaluated datasets. Confli-T5 shows the best performance (underlined values) on 23 out of the 30 evaluated samples in our experiments (excluding 0% samples lines). Moreover, Confli-T5 significantly outperforms the baselines in all augmentation factors when considering the average performance on all datasets (last line in Table III).

Although G3M is a powerful prompt-based baseline, it requires human inputs for prompt engineering, which may have hurt its performance. Tuning prompts on G3M is financially expensive once it implements GPT3 (not an open-source tool). On the other hand, EDA has a low complexity (and financial) cost. It produces more diverse data by using wordnet replacements and shuffling words. However, EDA ignores sentences' context and does not control the labels' consistency, which may hurt its performance.

Confli-T5 improved the classification performance observed when using the annotated samples only (0× column) in all sample sizes. It indicates that augmenting training data with Confli-T5 will improve (or at least not hurt) the performance in any sample size.

Confli-T5 does not rely on pre-existing annotated data to generate labeled samples. Although Confli-T5 is applicable only for text augmentation on conflict domains (or containing conflict topics), our model can generate data even without pre-existing annotated samples. Using the generated samples only for training the classifier produced good results. We believe that combining active learning with the Confli-T5 capability of generating labeled data from scratch may boost the quality of synthetic data with a small human input. Incorporating active learning mechanisms in Confli-T5 are part of our future work.

Confli-T5 continues improving the classification performance on large samples. Performance improvement on downstream text classification offered by augmentation techniques is usually more challenging on larger datasets because they tend to have a larger level of diversity. This effect is observed in Table III, where the performance gains using augmentation methods are larger on smaller samples. Still, our model improves the classification performance for 50% sample sizes, outperforming the baselines in five out of the six datasets.

V. Conclusions and Future Work

In this paper, we proposed Confli-T5, a prompt-based model which leverages the domain knowledge from CAMEO to generate synthetic text samples in conflict domain. Our model allows generating labeled data from the ground up, outperforming the baseline models in most of the tested scenarios.

TABLE III: Downstream classification performance (f1-scores): Confli-T5 vs. baselines.

Dataset	Samp. (%)	Augmentation Factor (Applied Over the Samples)																
		0×	1×				2×				3×				4×			
			EDA	**TMix**	**G3M**	**Ours**	**EDA**	**TMix**	**G3M**	**Ours**	**EDA**	**TMix**	**G3M**	**Ours**	**EDA**	**TMix**	**G3M**	**Ours**
CAMEO	**0%**	-	-	-	-	78.3	-	-	-	83.4	-	-	-	80.6	-	-	-	81.1
	1%	18.2	12.9	17.5	11.9	**28.1**	14.3	21.4	20.5	**29.5**	15.6	23.4	16.6	**38.4**	14.9	26.1	17.6	**47.8**
	5%	48.0	55.2	51.9	41.3	**60.7**	54.8	54.2	35.9	**73.2**	60.8	53.2	29.7	**71.5**	57.8	52.6	26.6	**77.0**
	10%	68.6	72.9	68.7	57.3	**75.7**	75.5	68.2	45.5	**80.8**	73.3	68.3	47.3	**82.3**	74.9	66.1	38.8	**82.8**
	25%	84.2	84.1	79.5	65.5	**86.1**	83.6	79.5	67.8	**86.7**	83.3	78.1	61.8	**88.1**	83.1	75.7	55.5	**88.5**
	50%	88.4	**88.7**	83.9	72.8	88.5	88.3	84.7	73.2	**90.0**	88.4	80.7	62.8	**89.8**	88.1	81.7	65.6	**90.7**
ACE05	**0%**	-	-	-	-	48.9	-	-	-	51.4	-	-	-	52.6	-	-	-	51.4
	1%	56.5	**60.6**	58.4	56.2	59.4	60.9	62.9	**67.2**	61.5	60.4	59.3	**61.6**	57.0	**61.6**	59.2	59.0	57.5
	5%	68.8	**73.2**	70.7	71.7	71.0	74.0	72.6	69.9	**74.3**	**75.3**	73.6	73.5	74.3	76.1	73.8	70.4	**77.0**
	10%	83.6	82.5	82.0	75.4	**85.5**	80.3	81.3	77.2	**85.9**	82.4	82.3	77.1	**85.6**	81.7	82.4	78.6	**85.4**
	25%	88.1	88.1	86.6	80.2	**88.4**	**88.0**	87.3	81.9	87.8	**88.8**	86.7	79.6	88.3	**88.9**	86.3	78.7	88.4
	50%	90.2	**89.9**	89.6	85.4	89.8	90.3	88.2	81.4	**90.5**	89.7	86.6	78.7	**89.8**	89.2	86.5	77.4	**89.8**
MAVEN	**0%**	-	-	-	-	61.8	-	-	-	58.2	-	-	-	59.0	-	-	-	57.0
	1%	64.4	40.3	48.0	37.9	**67.2**	41.5	56.1	49.0	**78.4**	44.1	66.1	69.9	**85.1**	46.4	54.2	32.6	**84.8**
	5%	80.2	85.5	83.6	83.8	**85.6**	86.7	83.2	78.1	**88.3**	87.7	84.3	80.9	**88.4**	**88.1**	77.1	58.7	87.9
	10%	88.6	**90.3**	87.6	79.9	90.1	90.3	88.0	80.5	**90.4**	**91.2**	84.4	70.2	89.8	**91.3**	82.6	65.4	89.8
	25%	90.3	90.3	90.0	88.6	**91.0**	90.5	78.4	53.0	**91.5**	**90.7**	77.5	51.5	90.2	90.7	84.8	72.2	**91.0**
	50%	91.2	91.2	89.7	85.6	**91.8**	91.5	81.5	59.6	**91.8**	90.9	86.2	74.6	**91.8**	91.4	82.0	61.0	**91.9**
Wiki	**0%**	-	-	-	-	66.6	-	-	-	66.7	-	-	-	65.7	-	-	-	64.0
	1%	46.4	45.4	47.6	**52.4**	45.1	53.2	**56.5**	53.8	46.0	50.8	54.5	**56.4**	43.1	51.4	52.9	**56.7**	51.0
	5%	60.1	63.7	64.5	65.1	**66.1**	58.9	64.0	62.7	**70.8**	62.9	64.6	61.9	**69.8**	63.8	65.4	61.5	**73.0**
	10%	72.1	67.7	69.7	70.0	**70.8**	67.4	68.5	62.6	**75.0**	65.7	69.8	68.4	**72.9**	66.9	69.4	66.5	**72.7**
	25%	74.9	71.7	73.1	68.8	**78.0**	73.7	75.0	70.9	**77.8**	75.0	73.9	67.3	**78.5**	73.9	73.1	66.5	**77.3**
	50%	78.4	**77.8**	76.4	71.4	77.8	76.6	76.9	71.7	**79.6**	77.1	76.0	70.3	**79.3**	75.3	75.2	70.0	**80.3**
GLOCON	**0%**	-	-	-	-	72.0	-	-	-	68.9	-	-	-	71.7	-	-	-	73.4
	1%	44.2	34.7	43.5	**55.7**	42.9	36.7	47.5	**60.7**	46.0	35.5	48.0	**64.0**	46.0	33.2	45.8	**61.6**	45.4
	5%	46.0	64.7	64.0	63.8	**66.4**	65.5	68.0	69.2	**71.2**	65.3	66.8	66.2	**71.7**	64.4	67.7	66.8	**74.5**
	10%	65.7	66.8	70.1	70.0	**71.0**	67.0	72.6	71.7	**76.4**	70.9	71.8	67.8	**75.6**	72.9	73.3	70.4	**76.2**
	25%	79.8	76.4	79.0	77.5	**80.8**	76.8	77.4	74.5	**80.9**	75.8	77.7	74.0	**81.8**	78.1	78.2	74.7	**80.4**
	50%	80.9	**82.6**	79.9	76.1	80.8	**82.6**	80.2	76.1	81.8	**82.2**	79.2	73.6	81.2	**84.1**	79.9	75.0	80.7
IndPol	**0%**	-	-	-	-	64.0	-	-	-	62.0	-	-	-	62.1	-	-	-	59.9
	1%	19.2	16.9	19.0	**23.2**	17.9	16.9	18.5	**20.3**	18.9	16.9	16.2	13.4	**19.3**	17.1	17.6	16.4	**20.9**
	5%	30.4	34.0	36.8	**42.3**	36.2	**44.1**	41.1	43.5	39.5	46.5	51.5	46.6	**61.7**	46.8	47.6	30.7	**69.5**
	10%	56.4	62.2	59.3	49.3	**65.3**	60.8	62.6	50.9	**73.7**	64.4	64.5	50.4	**77.7**	65.1	67.6	57.6	**78.2**
	25%	79.1	75.6	76.7	73.1	**81.1**	79.1	74.1	61.9	**80.2**	75.2	76.8	72.4	**82.3**	**79.7**	72.1	57.3	77.4
	50%	85.7	85.1	84.0	77.2	**87.8**	86.6	84.5	77.4	**88.6**	83.6	79.2	68.3	**85.5**	85.7	82.8	73.9	**86.9**
Average		67.6	67.7	67.7	64.3	**70.9**	68.5	68.5	62.3	**73.6**	69.0	68.7	61.9	**74.6**	69.4	68.0	58.8	**75.8**

Experiments have been repeated 10 times and presented results correspond to the mean of f1-score observed in the testing set of each data. Results in bold font indicate the best f1-score for each sample size and augmentation factor (**0.01**, **0.05** or $>$ **0.05** level of significance in t-test). To measure significance levels, we selected the highest p-value after comparing the best method versus all the others.

We believe that Confli-T5 can be successfully employed as a text augmentation method to support the advances in political and social sciences, promoting the management of global conflict. Future works can be summarized in three main directions: (i) develop active learning functionality to work with Confli-T5, (ii) develop a data augmentation module for named entity recognition using CAMEO, and (iii) explore multilingual functions for Confli-T5.

Acknowledgments

The research reported herein was supported in part by NSF awards DMS-1737978, DGE-2039542, OAC-1828467, OAC-1931541, and DGE-1906630, ONR awards N00014-17-1-2995 and N00014-20-1-2738, Army Research Office Contract No. W911NF2110032 and IBM faculty award (Research).

References

[1] J. Beieler and C. Norris, "Petrarch: Python engine for text resolution and related coding hierarchy," Available at https://github.com/openeventdata/petrarch (2020/05/15), 2014, unpublished Manuscript.

[2] C. Norris, P. Schrodt, and J. Beieler, "Petrarch2: Another event coding program," *Journal of Open Source Software*, vol. 2, no. 9, p. 133, 2017.

[3] J. Lu and J. Roy, "Universal petrarch: Language-agnostic political event coding using universal dependencies," Available at https://github.com/openeventdata/UniversalPetrarch (2020/05/22), 2017.

[4] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Neural Information Processing Systems (NIPS)*, pp. 5998–6008, 2017.

[5] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "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*. Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 4171–4186.

[6] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, "Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," *arXiv preprint arXiv:1910.13461*, 2019.

[7] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, P. J. Liu *et al.*, "Exploring the limits of transfer learning with a unified text-to-text transformer." *J. Mach. Learn. Res.*, vol. 21, no. 140, pp. 1–67, 2020.

[8] Y. Hu and L. Khan, "Uncertainty-aware reliable text classification," in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 2021, pp. 628–636.

[9] E. S. Parolin, L. Khan, J. Osorio, P. T. Brandt, V. D'Orazio, and J. Holmes, "3M-Transformers for Event Coding on Organized Crime Domain," in *2021 IEEE 8th International Conference on Data Science and Advanced Analytics (DSAA)*. IEEE, 2021, pp. 1–10.

[10] B. Büyüköz, A. Hürriyetoğlu, and A. Özgür, "Analyzing ELMo and DistilBERT on socio-political news classification," in *Proceedings of the Workshop on Automated Extraction of Socio-political Events from News 2020*. European Language Resources Association, pp. 9–18.

[11] E. S. Parolin, Y. Hu, L. Khan, J. Osorio, P. T. Brandt, and V. D'Orazio, "CoMe-KE: A new transformers based approach for knowledge extraction in conflict and mediation domain," in *2021 IEEE International Conference on Big Data (Big Data)*. IEEE, 2021, pp. 1449–1459.

[12] Y. Hu, M. S. Hosseini, E. S. Parolin, J. Osorio, L. Khan, P. Brandt, and V. D'Orazio, "Conflibert: A pre-trained language model for political conflict and violence," in *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2022.

[13] E. S. Parolin, M. Hosseini, Y. Hu, L. Khan, J. Osorio, P. T. Brandt, and V. D'Orazio, "Multi-CoPED: A Multilingual Multi-Task Approach for Coding Political Event Data on Conflict and Mediation Domain," in *Proceedings of the 2022 AAAI/ACM Conference on AI, Ethics, and Society*, 2022.

[14] T. Schick and H. Schütze, "Exploiting cloze-questions for few-shot text classification and natural language inference," in *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, 2021, pp. 255–269.

[15] T. Shin, Y. Razeghi, R. L. Logan IV, E. Wallace, and S. Singh, "Autoprompt: Eliciting knowledge from language models with automatically generated prompts," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 4222–4235.

[16] Z. Jiang, F. F. Xu, J. Araki, and G. Neubig, "How can we know what language models know?" *Transactions of the Association for Computational Linguistics*, vol. 8, pp. 423–438, 2020.

[17] K. M. Yoo, D. Park, J. Kang, S.-W. Lee, and W. Park, "GPT3Mix: Leveraging large-scale language models for text augmentation," in *Findings of the Association for Computational Linguistics: EMNLP 2021*. Association for Computational Linguistics, 2021, pp. 2225–2239.

[18] Y. Hu, Y. Lin, E. S. Parolin, L. Khan, and K. Hamlen, "Controllable fake document infilling for cyber deception," *arXiv preprint arXiv:2210.09917*, 2022.

[19] J. Wei and K. Zou, "EDA: Easy data augmentation techniques for boosting performance on text classification tasks," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing(EMNLP)*. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 6383–6389.

[20] X. Zhang, J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification," *Advances in neural information processing systems*, vol. 28, 2015.

[21] M. Fadaee, A. Bisazza, and C. Monz, "Data augmentation for low-resource neural machine translation," in *In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, 2017.

[22] Q. Xie, Z. Dai, E. Hovy, T. Luong, and Q. Le, "Unsupervised data augmentation for consistency training," *Advances in Neural Information Processing Systems*, vol. 33, pp. 6256–6268, 2020.

[23] J. Chen, Y. Wu, and D. Yang, "Semi-supervised models via data augmentation for classifying interactive affective responses," *Workshop On Affective Content Analysis, The ThirtyFourth AAAI Conference on Artificial Intelligence*, 2020.

[24] V. Kumar, A. Choudhary, and E. Cho, "Data augmentation using pre-trained transformer models," in *Proceedings of the 2nd Workshop on Life-long Learning for Spoken Language Systems*. Suzhou, China: Association for Computational Linguistics, Dec. 2020, pp. 18–26.

[25] C. Xia, C. Xiong, P. Yu, and R. Socher, "Composed variational natural language generation for few-shot intents," *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 3379–3388, 2020.

[26] C. Xia, C. Zhang, H. Nguyen, J. Zhang, and P. Yu, "Cg-bert: Conditional text generation with bert for generalized few-shot intent detection," *arXiv preprint arXiv:2004.01881*, 2020.

[27] Y. Hou, Y. Liu, W. Che, and T. Liu, "Sequence-to-sequence data augmentation for dialogue language understanding," *Proceedings of the 27th International Conference on Computational Linguistics*, 2018.

[28] K. M. Yoo, Y. Shin, and S.-g. Lee, "Data augmentation for spoken language understanding via joint variational generation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019, pp. 7402–7409.

[29] N. Ng, K. Cho, and M. Ghassemi, "SSMBA: Self-supervised manifold based data augmentation for improving out-of-domain robustness," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, 2020, pp. 1268–1283.

[30] D. Guo, Y. Kim, and A. Rush, "Sequence-level mixed sample data augmentation," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, Nov. 2020, pp. 5547–5552.

[31] L. Sun, C. Xia, W. Yin, T. Liang, P. Yu, and L. He, "Mixup-transformer: Dynamic data augmentation for NLP tasks," in *Proceedings of the 28th International Conference on Computational Linguistics*. Barcelona, Spain (Online): International Committee on Computational Linguistics, Dec. 2020, pp. 3436–3440.

[32] J. Chen, Z. Yang, and D. Yang, "MixText: Linguistically-informed interpolation of hidden space for semi-supervised text classification," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, Jul. 2020, pp. 2147–2157.

[33] B. O'Connor, B. M. Stewart, and N. A. Smith, "Learning to extract international relations from political context," *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, 2013.

[34] A. Hanna, "Mpeds: Automating the generation of protest event data," 2017, unpublished Manuscript.

[35] J. Osorio, A. Reyes, A. Beltrán, and A. Ahmadzai, "Supervised event coding from text written in Arabic: Introducing hadath," in *Proceedings of the Workshop on Automated Extraction of Socio-political Events from News 2020*. European Language Resources Association.

[36] M. Solaimani, S. Salam, L. Khan, P. T. Brandt, and V. D'Orazio, "Repair: Recommend political actors in real-time from news websites," *Proceedings of the International Conference on Big Data (Big Data)*, pp. 1333–1340, 2017.

[37] J. Beieler, "Generating politically-relevant event data," *In Proceedings of the First Workshop on NLP and Computational Social Science*, 2016.

[38] G. Glavaš, F. Nanni, and S. P. Ponzetto, "Cross-lingual classification of topics in political texts," in *Proceedings of the Second Workshop on NLP and Computational Social Science*. Vancouver, Canada: Association for Computational Linguistics, Aug. 2017, pp. 42–46.

[39] F. K. Örs, S. Yeniterzi, and R. Yeniterzi, "Event clustering within news articles," in *Proceedings of the Workshop on Automated Extraction of Socio-political Events from News 2020*. European Language Resources Association, pp. 63–68.

[40] E. S. Parolin, L. Khan, J. Osorio, V. D'Orazio, P. Brandt, and J. Holmes, "Hanke: Hierarchical attention networks for knowledge extraction in political science domain," *IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, 2020.

[41] A. Holtzman, J. Buys, L. Du, M. Forbes, and Y. Choi, "The curious case of neural text degeneration," in *International Conference on Learning Representations*, 2020.

[42] A. Fan, M. Lewis, and Y. Dauphin, "Hierarchical neural story generation," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2018, pp. 889–898.

[43] A. Holtzman, J. Buys, M. Forbes, A. Bosselut, D. Golub, and Y. Choi, "Learning to write with cooperative discriminators," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2018, pp. 1638–1649.

[44] X. Wang, Z. Wang, X. Han, W. Jiang, R. Han, Z. Liu, J. Li, P. Li, Y. Lin, and J. Zhou, "MAVEN: A Massive General Domain Event Detection Dataset," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, Nov. 2020, pp. 1652–1671.

[45] S. Li, H. Ji, and J. Han, "Document-level event argument extraction by conditional generation," in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Online: Association for Computational Linguistics, Jun. 2021, pp. 894–908.

[46] A. Hürriyetoğlu, E. Yörük, D. Yüret, Ç. Yoltar, B. Gürel, F. Duruşan, and O. Mutlu, "A task set proposal for automatic protest information collection across multiple countries," in *European Conference on Information Retrieval*. Springer, 2019, pp. 316–323.

[47] A. Halterman, K. Keith, S. Sarwar, and B. O'Connor, "Corpus-level evaluation for event qa: The indiapoliceevents corpus covering the 2002 gujarat violence," in *Findings of the Association for Computational Linguistics (ACL-IJCNLP) 2021*, pp. 4240–4253.