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

***Abstract*—Recent advances in natural language processing (NLP) and Big Data technologies have been crucial for scientists analyzing political unrest and violence, preventing harm and promoting the management of global conflict. Government agencies and public security organizations have heavily invested on deep learning based applications to study conflicts and political violence globally. However, such applications involving text classification, information extraction and other NLP related tasks require extensive human efforts on annotating/labeling texts. While limited labeled data may drastically hurt the models' performance (over-fitting), large demands on annotation task may turn real-world applications impracticable. To address this problem, we propose Confli-T5, a prompt-based method which leverages the domain knowledge from existing political science ontology to generate synthetic but realistic labeled text samples involving 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 textual 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 very large unstructured data (news articles) are crucial tasks to monitoring, understanding, and predicting the dynamics of social unrest, political violence, and armed conflict worldwide.

Along the past two decades, political scientists and computational linguistics have been 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 traditional NLP tasks in political science and conflict domains.

Recent advances in deep-learning and computational linguistics have been pushing political science scholars to focus their efforts in the second direction. Previous efforts employing transformer-based [4] pre-trained language models (PLMs) (e.g., BERT [5]) have shown successful results in several political science subareas, such as organized crime [6], protests [7], and general conflict and mediation topics [8]–[11]. The continuous advances on this direction are crucial for scientists analyzing political unrest and violence, preventing harm, and promoting the management of global conflict.

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

To address this problem, we propose Confli-T5, a pipeline model for generating synthetic text samples in 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 [12] to generate synthetic labeled data for text classification. Our method differs from previous augmentation models by dispensing human inputs on prompt engineering, and by 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 purposes. Second, we design our model to allow generating labeled textual samples, not requiring pre-existing labeled data (as the other baseline models do). Third,

[1] https://drive.google.com/drive/folders/1VF35tdEsHuzvMCLdoP-0tGhg7hjIbguG?usp=sharing

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 the empirical results of existing text augmentation methods with ours.

II. Preliminaries

A. Related Work

Pre-trained Language Models (PLM). Deep neural networks based on self-attention structures introduced by transformer [4] stretched the performance boundaries of language modeling in NLP community. Transformer-based models such as BERT [5], DistilBERT [13] and BART [14] allow transfer learning through pre-train and fine-tune frameworks, reaching state-of-the-art results in all traditional NLP tasks. Specifically, T5 [12] is a unified model that works in a sequence-to-sequence fashion by converting text-based language problems into a text-to-text format.

Prompt-based Learning. Traditionally, prompt engineering in NLP consists on embedding the description of the task to be solved as part of the input sequence. In practice, prompt methods convert one or more tasks to a prompt-based dataset (by adding prefixes associated to tasks) to learn language models on those tasks. Recent works have been focusing on employing PLMs for zero/few-shot learning through prompt-based mechanisms [8], [15]–[20]. 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 has been a useful technique given the extensive costs (time, money and expertise) associated to 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 [21], [22] and paraphrasing technique based on back-translation [23]–[25]. Other works explored large-scale language models by prepending the existing class labels to input sequences [26], perturbing latent spaces [27]–[30], or employing masked language models as denoising autoencoder [31] to generate synthetic data. Recent works [20], [32]–[34] have introduced mix-up based approaches for augmentation by mixing pre-existing samples (or interpolating them in their corresponding hidden space) 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 labels associated to them (through our *double random sampling* method). By maintaining consistency property, we mitigate noisy data points and improve the performance on 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. Most previous works for coding event data are based on pattern-matching approaches [1]–[3], [35], [36], usually supported by large repositories or ontologies. Recent works have successfully applied transformer-based neural networks for coding events. Specifically, [11] have empirically shown the significant superiority of BERT implementation for coding events over pattern-matching approaches. Other studies have concentrated on conflict event detection employing classical machine learning [36]–[39] and deep learning [40]–[44] techniques. Further, [10] pre-trained a PLM to generally attend NLP tasks on conflict domain.

Next, we describe relevant details about CAMEO, which is 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 ($\approx$ 14K entries) and **actor** dictionaries ($\approx$ 67K entries).

The action-pattern repository stores verbal patterns (resembling regular expressions) associated to categories of political interactions (known as CAMEO codes). Despite the high granularity of event types offered by CAMEO (more than 200 action codes), conflict scholars traditionally use a higher level of categories, grouping the original types into twenty (rootcodes) or five classes (pentacodes), as summarized in Table I and detailed in CAMEO **codebook**[2].

TABLE I: Rootcodes and pentacodes descriptions.

CAMEO Codes	Rootcodes	Pentacodes
010 - 019	01- Make Public Statement	0- Make a Statement
020 - 028	02- Appeal	0- Make a Statement
030 - 039	03- Express Intent to Cooperate	1- Verbal Cooperation
040 - 046	04- Consult	1- Verbal Cooperation
050 - 057	05- Engage in Diplomatic Cooperation	1- Verbal Cooperation
060 - 064	06- Engage in Material Cooperation	2- Material Cooperation
070 - 075	07- Provide Aid	2- Material Cooperation
080 - 0874	08- Yield	2- Material Cooperation
090 - 094	09- Investigate	3- Verbal Conflict
100 - 108	10- Demand	3- Verbal Conflict
110 - 116	11- Disapprove	3- Verbal Conflict
120 - 129	12- Reject	3- Verbal Conflict
130 - 139	13- Threaten	3- Verbal Conflict
140 - 1454	14- Protest	4- Material Conflict
150 - 155	15- Exhibit Force Posture	4- Material Conflict
160 - 1663	16- Reduce Relations	3- Verbal Conflict
170 - 176	17- Coerce	4- Material Conflict
180 - 186	18- Assault	4- Material Conflict
190 - 196	19- Fight	4- Material Conflict
200 - 2042	20- Unconventional Mass Violence	4- Material Conflict

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 (see Table I). In this example,

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

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.

CAMEO is basically a static ontology where the knowledge rests. As aforementioned in 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

In this section we describe the components of our pipeline model 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 is employed to generate synthetic labeled texts. Then, BART works as a natural language inference (NLI) parser to improve the quality of the generated data, which finally will serve as augmented data to train a supervised model for a downstream task. In this paper, we focus on text generation for classification purpose, leaving the analysis on other tasks (e.g., named entity recognition) as future work.

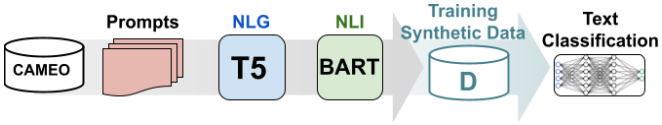


Fig. 1: Diagram of text augmentation with Confli-T5.

A. *CAMEO-based Prompts*

Before describing the procedure to construct the prompts, we formalize the following rules previously discussed in Subsection II-B: every political actor ρ stored in CAMEO is associate to an actor code $code_p(\rho)$; as well as a 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 prompts' template, showing three real examples to demonstrate the whole procedure. The template of our prompts is composed by three parts: action-pattern ν, actors (source ρ_{src} represented by $ and target ρ_{tgt} represented by +) and prefix. Specifically, the prefix consists of a brief description associated to the action code $code_v(\nu)$. Such descriptions (used as prefixes) are extracted from CAMEO codebook.

Our procedure depicted in Fig. 2 first randomly selects an action-pattern ν from 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 $code_{src}$ 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 to code $code_{src}$ appears as source of any action ν with code $code_v(\nu)$. The actor ρ_{tgt} is selected in the same manner, using $P(tgt = code_{src} \mid code_v(\nu))$ instead. These conditional probabilities were pre-computed based on statistics observed on dataset available from previous study [45] (1,920,174 real-world sentences collected from 400 news agencies spread around the world). 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, the components aforementioned are put 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 words from action-pattern to indicate the places where the NLG model will fill up. The CAMEO codes $code_v(\nu)$ associated to action-patterns ν will later serve to indicate the labels for the prompts using ν.

B. *Double Random Sampling Strategy*

As depicted in Fig. 1, T5 is employed as NLG model for text infilling on CAMEO-based prompts (as exemplified in Fig. 2). Following, BART will work as 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.

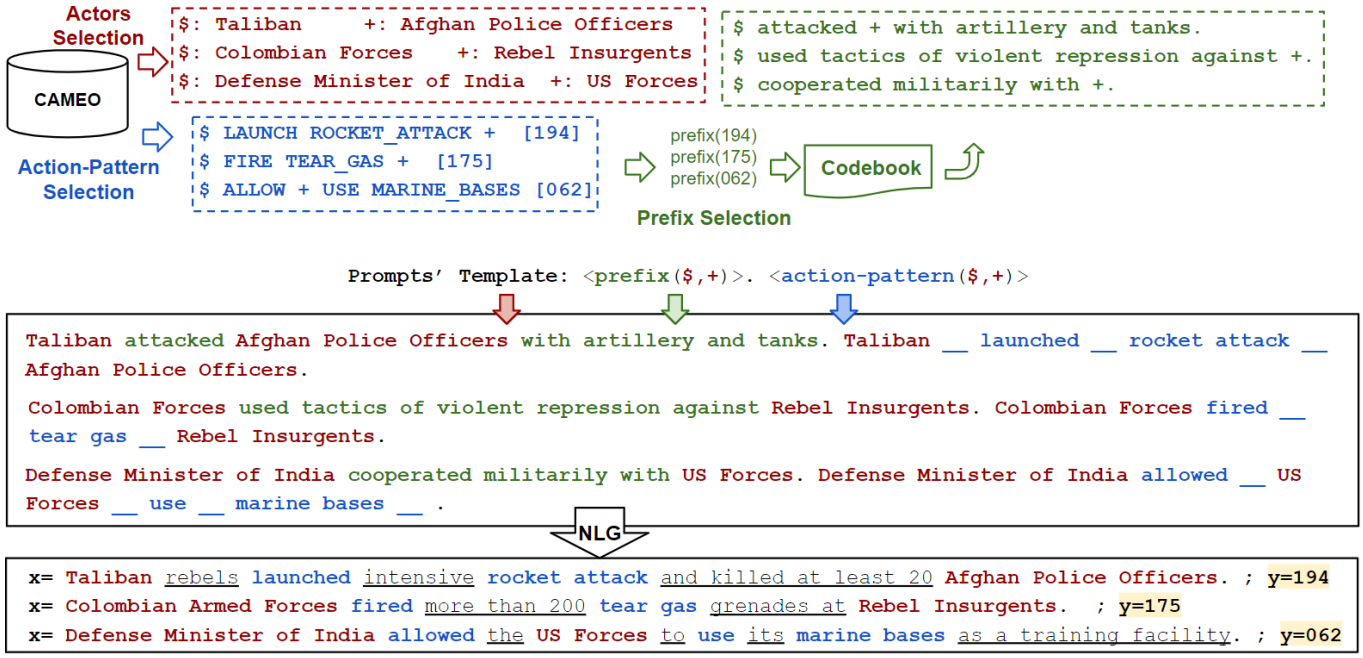


Fig. 2: Actors and action-patterns are randomly selected from CAMEO ontology. The prompt prefixes are selected based on the action-codes. Actions, actors and prefixes will then fill the prompts' template to construct the prompts (including blanks). The prompts will feed the NLG model, which in turn fills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

In our implementation, we use *nucleus sampling* [46] as 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 taking into consideration 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 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, by making the distribution less random (skewing the distribution towards high probability events) and 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 which determines whether a *hypothesis* is true (entailment), false (contradiction), or undetermined (neutral) given a *premise*. Both text sequences for premise and hypothesis are given as input to the model. Confli-T5 implements the transformer-based BART for NLI as zero-shot mechanism to verify whether the generated texts are consistent to 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 hypothesis and the text generated from the action-pattern as premise. Given the (premise, hypothesis) pair as input, we use BART entailment score to identify incoherent or inconsistent generated samples.

Table II shows some 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 to 4, the generated texts are consistent to their corresponding prefixes, with the high entailment scores reinforcing such consistency. On the other hand, examples from 5 to 7 show low entailment scores, indicating either lack of consistency between generated text and prefixes (Ex.IDs 6 and 7) or lack of coherence in the generated text (Ex.ID 5).

However, we noted that 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 3 and 4). While searching examples with slightly lower entailment scores (Ex.IDs 9 to 11), we observed that these generated samples add more tokens over the original patterns, yet keeping consistency and coherency.

It seems beneficial to get rid of the samples with low NLI scores to avoid noisy examples in the training synthetic data.

TABLE II: 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	Defense Minister of India allowed the US Forces to use its marine bases as a training facility.	Defense Minister of India cooperated militarily with US Forces.	$ **ALLOW** + USE MARINE BASES	062	0.9915
2	Colombian Armed Forces fired more than 200 tear gas grenades at Rebel Insurgents.	Colombian Armed Forces used tactics of violent repression against Rebel Insurgents.	$ **FIRE** TEAR_GAS +	175	0.9994
3	Italy lifted ban on trade with Cuba in 2009.	Italy eased economic sanctions on Cuba.	$ **LIFT** BAN ON TRADE WITH +	085	0.9971
4	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
5	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
6	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
7	U.N. efforts have failed to prevent the attack on Germany.	U.N. supported Germany.	$ EFFORTS TO **PREVENT** ATTACK +	053	0.0077
8	U.S. urged Beijing to deploy its armed forces in the region to protect Taiwan.	U.S. supported Taiwan.	$ **URGE** TO DEPLOY FORCES TO PROTECT +	053	0.8236
9	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
10	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
11	Demonstrators protested against racist remark by California Governor Jerry Brown.	Demonstrators engaged in civilian demonstrations to protest against California Governor.	$ **PROTEST** REMARK BY +	140	0.9629

Besides, searching for distinct and more natural 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 [47]–[49]) to select the generated sentences from $\tilde{D}$.

Top-q sampling first filters the subset of sentences $Q \subset \tilde{D}$ such that entailment score is higher than a threshold q. From Q, it constructs the **training synthetic 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

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

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 from Table I), 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 are useful not only for providing a context (W_0 in Eq. 1) but also 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 in Algorithm 1 all the steps previously discussed. 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 Λ (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** Λ *is not None* **then** $D \leftarrow \Lambda$
6 **else** $D \leftarrow \{\emptyset\}$
7 **foreach** *(code* τ*, probability* $P\tau$*) in CAMEO2distr.items()* **do**
8 y ← CAMEO2labels[τ]
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

Smaller portions of labeled data can be added to final training set D through Λ. Furthermore, additional data out of conflict domain (e.g., sports, technology or religion) can also be included to 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 training labels. Text x appended to D in line 12 is composed by generated texts only, discarding prefixes and NLI scores.

IV. Experiments and Results

A. Setup

To conduct the experiments presented in this paper, we used a computer with one Quadro RTX 8000 GPU. We run 10 rounds of training process for each experimented model and report the averaged results observed on testing set. In each round, we generate different train/validation splits (85%/15% over training data) and randomly initialize the model based on the seed assigned for that round. We train our models over 20 epochs and the best model of each round is selected based on F1-scores observed on their corresponding validation splits. We use 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 utilize the same full synthetic corpus D of size $|D|$=408,000 and explore it using top-q search with different topics (codes in Table I), as expressed in Algorithm 1.

As pre-trained language models, we used *t5-large* for T5 and *bart-large-mnli* for zero-shot BART. As transformer-based network for training the models with synthetic data, we used *bert-base-uncased* and *ConfliBERT-cont-uncased* [3] [10].

For a more comprehensive evaluation, we selected three augmentation methods using completely different approaches as baselines. EDA [21] applies simple operations such as synonym replacement, random insertion, random swap, and random deletion to augment text. TMix [34] creates large amount of augmented training samples by interpolating text hidden space in BERT model. Finally, GPT3Mix [20] (G3M in experiments) is a prompt-based generation method that uses pseudo-labeling to generate text samples with their soft-labels. We use the hyperparameters reported by the authors. The data splits, number of seeds and reporting approach were exactly the same for all the models evaluated in this section.

B. Datasets

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

Conflict and Mediation Observations (CAMEO) [11] is a sentence level data, following the industry standard schema for event extraction in political science (see II-B). Data points are annotated with the actions (pentacodes) occurring in the sentences. In our experiments, we remove the records associated to pentacode *0-Make a Statement* (see Table I) to concentrate our analysis on conflict and mediation related topics.

Automatic Content Extraction 2005 (ACE05) is a standard dataset widely used on event extraction and NLP researches. Overall, it annotates 33 event types, including conflict-related subjects (Attack and Demonstrate labels), which correspond to approximately 30% of the total annotated events. Once political and social scientists are often interested in extracting conflict-related events from large corpora, we understand ACE05 is an appropriate data for evaluating whether synthetic data from Confli-T5 can train supervised models to perform such task. For our experiments, we select distinct sentences, marking those containing conflict-related events as 1 and the remaining records as 0 for binary classification task.

Massive Event Detection (MAVEN) [50] is another standard dataset for event extraction, which annotates 168 event types, including military, civil and terrorist related conflicts. In our experiments, we utilize the topic labels of documents to split the original document-level data in three conflict categories for text classification, as described in Table III.

WikiEvents (Wiki) [51] is a document-level event extraction dataset containing 50 event types, including conflict related categories such as violent attack and demonstration. For our experiments, we collect the sentences containing any conflict related event with flag 1 and the remaining sentences as 0 (see Table III), similarly as we did for ACE05.

Global Contention Politics (GLOCON) [52] 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 utilize GLOCON data following exactly the same format used in previous work [10].

India Police Events (IndPol) [53] contains news sentences (in English language) from Times of India articles reporting police activity events during a period of widespread Hindu-Muslim violence in Gujarat (March 2002). The sentences are annotated in multi-label fashion considering four categories of police activity: kill, arrest, fail to act and force. In our experiments, we remove the data points either containing no police activity events or containing more than one event.

TABLE III: 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 to 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.Conflict,Mil. Attack, Mil.Operation	15 and 19	1
		Civ.Attack, Civ.Conflict, Terrorist Attack	14	2
		Others	4 and 5	0
Wiki	1,582/396	Conflict (Attack, Demonstrate, Defeat)	14, 18 and 19	1
		Others	3, 4 and 7	0
GLOCON	1,548/388	Protest	14	1
		No Protest	1 to 8	0
IndPol	555/140	Kill	(1823,185, 186,202)	0
		Arrest	(173)	1
		Fail to Act	5 and 12	2
		Force	(170 to 173, 175,180, 190 to 193)	3

[3] https://huggingface.co/snowood1/ConfliBERT-scr-uncased

TABLE IV: 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**

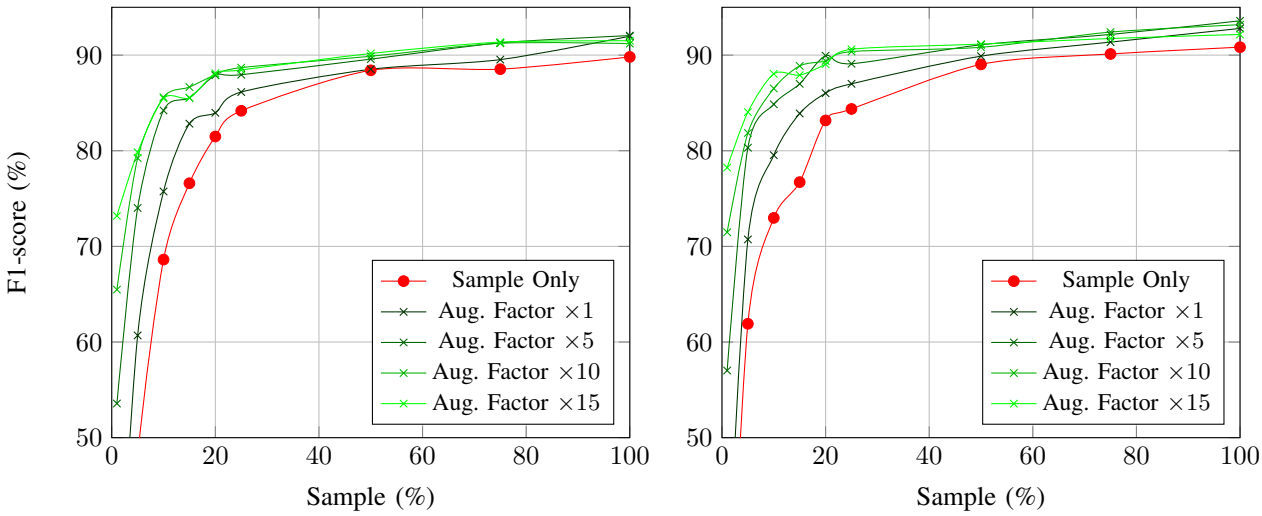
Experiments have been repeated 10 times and presented results correspond to the mean of f1-score observe in testing set of each data. Results in bold font indicate the best f1-score for each sample size and augmentation factor (**0.01**, **0.05** or $>$ **0.05** level of significance in t-test). To measure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

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

C. Data Augmentation Experiments

Traditionally, text augmentation methods require a pre-existing portion of annotated text to augment from it. So, for our experiments, we randomly sample the existing training data into smaller portions (e.g., 1%, 5%, 10%, 25%, 50% of the original size) and assume that these samples are the pre-existing annotated data available. We apply the augmentation methods to synthesize data of different sizes, increasing the pre-existing sample by an *augmentation factor* (e.g., $1\times$ or $2\times$ of the sample size). Finally, we train the downstream classification model BERT using the pre-existing plus the synthetic samples as training data and measure the classification performance using the original test sets. We can add pre-existing data and control the augmentation factor in Confli-T5 through the inputs Λ and N in Algorithm 1, respectively.

Table IV shows the f1-scores observed on downstream classification over the six datasets (see IV-B), considering 20 possible scenarios (5 sample sizes $\times$ 4 augmentation factors). The values under the column $0\times$ 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



(a) Confli-T5 + BERT

(b) Confli-T5 + Confli-BERT

Fig. 3: F1-score on CAMEO dataset varying sample size and augmentation factors

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 IV.

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 observed performance (underlined values) on 23 out of the 30 evaluated samples on 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 the Table IV).

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 and produces more diverse data by using wordnet replacements and shuffling words. However, EDA ignores sentences context and does not control the label consistency, which certainly may have hurt its performance.

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

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 domain (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 mechanism in Confli-T5 are part of our future work.

Confli-T5 continues improving the performance on downstream classification 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 IV, 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.

To better illustrate the gains curve provided by Confli-T5, we stretch the sample sizes and augmentation factors to evaluate the performance on CAMEO data in Fig. 3a. The chart shows that the performance can still be improved with synthetic data even when using the whole dataset as part of the training set. Same findings were observed for the other five dataset evaluated in our experiments (plots are suppressed due to space constraints). Chart in Fig. 3b shows similar analysis, but instead of fine-tuning common BERT, we do it over ConfliBERT [10]. Once this model was pre-trained with conflict domain data only, it is expected to learn faster/better, requiring lower volume of conflict-related data. Still, the conclusions are the same as those observed in Fig. 3a, indicating that Confli-T5 can improve the results on all experimented sample sizes even when using Confli-BERT.

D. Parameter Study

The two most important hyperparameters to be tuned in our method are p and q, for nucleus and top-q sampling, respectively. Based on previous studies [46], varying p affects the fluency and diversity of the generated text. However, for

our application, p can not control the consistency between the generated text and the label associated to it. On the other hand, q impacts in both diversity and consistency: $q \to 0$ increases the diversity and decreases the labels consistency, while $q \to 1$ will behave in the opposite direction. Once ensuring label consistency is a crucial aspect for our application, we focus on analyzing the effect of varying q while keeping p=0.9 (which produced better results on our preliminary validations).

Table V shows the classification performance on CAMEO dataset, synthesizing data of different sizes of N using three model configurations. The first, called *Uniform*, implements Confli-T5 without NLI layer, uniformly selecting N entries from $\tilde{D}$. The second, called *Greedy*, implements Confli-T5 with NLI layer, selecting the top N samples with largest NLI scores from $\tilde{D}$. The third one is Confli-T5 using top-q sampling with different values for hyperparameter q.

TABLE V: Uniform vs. Greedy vs. Top-q Sampling.

N	Unif.	Greedy	Top-q Sampling			
			q=0.975	q=0.95	q=0.90	q=0.85
1,000	74.76	67.14	77.12	77.26	76.02	**77.90**
2,500	80.32	74.61	81.33	81.59	**82.58**	79.92
5,000	80.93	82.20	**83.30**	81.82	79.76	82.86
10,000	80.28	81.97	**83.26**	82.79	82.77	83.10
15,000	81.67	82.58	83.56	82.68	**84.69**	83.93
20,000	81.98	82.50	**83.42**	83.40	82.74	82.52
30,000	80.09	83.63	**85.12**	84.65	84.67	83.03
40,000	82.43	82.92	83.55	83.69	83.90	**84.07**
50,000	81.82	83.48	84.49	82.10	**84.94**	84.37

Results in **bold** font indicate the best f1-score for each size N.

We first note that adding NLI layer in Confli-T5 improves the quality of the generated data. The classification performance using greedy strategy is better than when we disregard the NLI layer (Unif.) for all sizes of N, except for the smallest sets ($1,000$ and $2,500$). It occurs because these small sets (with largest NLI scores) are composed by sentences that barely reproduce the CAMEO action-patterns (as discussed in Subsection III-B and Table II). Such sentences tend to be too semantically close to the codebook prefixes, making the synthetic training data too homogeneous. As a result, the low level of diversity introduced in this data will preclude the classification models trained over them to generalize well.

To increase the diversity in synthetic data, we use top-q sampling, which controls labels consistency and simultaneously allows certain level of randomness. Results on Table V show an overall improvement on downstream classification when using top-q, even for smaller sizes of N. In particular, q=0.975 consistently produced better results than uniform and greedy sampling, outperforming the other values for q in four out of the nine tested sizes for N. For this reason, we used q=0.975 for all experiments presented in this section.

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.

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 data augmentation module for named entity recognition using CAMEO, and (iii) explore multi-lingual function for Confli-T5.

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] 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.

[7] 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*. Marseille, France: European Language Resources Association (ELRA), May 2020, pp. 9–18.

[8] 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.

[9] F. Olsson, M. Sahlgren, F. ben Abdesslem, A. Ekgren, and K. Eck, "Text categorization for conflict event annotation," in *Proceedings of the Workshop on Automated Extraction of Socio-political Events from News 2020*. Marseille, France: European Language Resources Association (ELRA), May 2020, pp. 19–25.

[10] Y. Hu, M. S. Hosseini, E. Skorupa 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.

[11] 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.

[12] 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.

[13] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter," *arXiv preprint arXiv:1910.01108*, 2019.

[14] 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.

[15] T. Schick and H. Schütze, "Exploiting cloze questions for few shot text classification and natural language inference," *arXiv preprint arXiv:2001.07676*, 2020.

[16] T. Shin, Y. Razeghi, R. L. Logan IV, E. Wallace, and S. Singh, "Autoprompt: Eliciting knowledge from language models with automatically generated prompts," *arXiv preprint arXiv:2010.15980*, 2020.

[17] 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.

[18] L. Reynolds and K. McDonell, "Prompt programming for large language models: Beyond the few-shot paradigm," in *Extended Abstracts of the Conference on Human Factors in Computing Systems*, 2021, pp. 1–7.

[19] Z. Zhao, E. Wallace, S. Feng, D. Klein, and S. Singh, "Calibrate before use: Improving few-shot performance of language models," in *International Conference on Machine Learning*. PMLR, 2021, pp. 12 697–12 706.

[20] 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.

[21] 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 and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 6383–6389.

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

[23] 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, pp. 567–573.

[24] 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.

[25] 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.

[26] 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.

[27] 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.

[28] 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.

[29] 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*, pp. 1234–1245, 2018.

[30] 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.

[31] 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.

[32] 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.

[33] 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.

[34] 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.

[35] J. Osorio and A. Reyes, "Supervised Event Coding From Text Written in Spanish: Introducing Eventus ID," *Social Science Computer Review*, vol. 35, no. 3, pp. 406–416, 2017.

[36] 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*. Marseille, France: European Language Resources Association (ELRA), May 2020, pp. 49–56.

[37] 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*, vol. 1, pp. 1094–1104, 2013.

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

[39] 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.

[40] J. Beieler, "Generating politically-relevant event data," *In Proceedings of the First Workshop on NLP and Computational Social Science*, pp. 37–42, 2016.

[41] B. Radford, "Multitask models for supervised protest detection in texts," 2019, unpublished Manuscript.

[42] 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.

[43] 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*. Marseille, France: European Language Resources Association (ELRA), May 2020, pp. 63–68.

[44] B. Radford, "Seeing the forest and the trees: Detection and cross-document coreference resolution of militarized interstate disputes," in *Proceedings of the Workshop on Automated Extraction of Socio-political Events from News 2020*. Marseille, France: European Language Resources Association (ELRA), May 2020, pp. 35–41.

[45] 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.

[46] 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.

[47] 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.

[48] 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.

[49] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners."

[50] 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.

[51] 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.

[52] 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.

[53] 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*, 2021, pp. 4240–4253.