Hybrid Knowledge Engineering Leveraging a Robust ML Framework to Produce an Assassination Dataset

Abigail Sticha and **Ernesto Verdeja** and **Paul Brenner**
University of Notre Dame
Notre Dame, IN 46556
{asticha, everdeja, paul.r.brenner}@nd.edu

Abstract

Social and political researchers require robust event datasets to conduct data-driven analysis, an example being the need for trigger event datasets to analyze under what conditions and in what patterns certain trigger-type events increase the probability of mass killings. Fortunately, NLP and ML can be leveraged to create these robust datasets. In this paper we (i) outline a robust ML framework that prioritizes understandability through visualizations and generalizability through the ability to implement different ML algorithms, (ii) perform a comparative analysis of these ML tools within the framework for the coup trigger, (iii) leverage our ML framework along with a unique combination of NLP tools, such as NER and knowledge graphs, to produce a dataset for the the assassination trigger, and (iv) make this comprehensive, consolidated, and cohesive assassination dataset publicly available to provide temporal data for understanding political violence as well as training data for further sociopolitical research.

1 Introduction

Peace and conflict researchers have identified several large-scale structural conditions that make state-led mass killings more likely, such as ongoing political instability or histories of state violence against vulnerable groups (Verdeja, 2016). However, the timing of mass killing onset is less understood. Burley et al. (2020) identifies nine potential triggering events for state mass killings, such as coups and assassinations, but before socio-political researchers can conduct systematic analysis to examine whether, and if so when, certain patterns of trigger-type events actually increase the probability of mass killings, it is necessary for political researchers to obtain political event datasets for each of these potential triggering events.

Processing the massive amount of information in available data in order to create socio-policial event (SPE) datasets for events such as the triggers described above takes extensive time, money, and human power. Fortunately, natural language processing (NLP) and related machine learning (ML) tools can be harnessed to classify the rapidly growing, but often poorly structured and unlabeled, data as to whether they contain an event or not. ML classification tools have been increasing combined with other NLP tools such as Named Entity Recognition (NER) and Knowledge Graphs (KGs) to engineer these datasets. Although these ML and NLP tools have become more robust, it is important for the AI research community to acknowledge that each tool comes with limitations and a scope of use. With this in mind, our project seeks to uniquely leverage a combination of these tools in order to mitigate their drawbacks to create an SPE dataset.

The most cited challenge for political event extraction is small labeled training datasets (Büyüköz et al., 2020; Ramrakhiyani et al., 2021; Caselli et al., 2021) which become an issue when working with ML classification algorithms. Therefore, our first task is to provide a clear, efficient, and accessible machine learning framework that future social scientists may utilize when implementing NLP-focused algorithms to classify large quantities of text documents given a small labeled training dataset. We prioritize a framework that is reproducible, understandable, and generalizable by both including essential visualizations of the input data and results and structuring the framework in such a way that fellow researchers can implement different ML algorithms, such as support vector machines (SVMs) or bidirectional encoder representations from transformers (BERT). We demonstrate that different ML algorithms are most suitable for a given optimization problem by performing a thorough comparative analysis of these different ML algorithms for the coup trigger in the process of refining our framework.

After explaining our ML framework, we demon-

Proceedings of the 5th Workshop on Challenges and Applications of Automated Extraction of Socio-political Events from Text (CASE), pages 106 - 116
December 7-8, 2022 ©2022 Association for Computational Linguistics

strate how we implement this framework to create a dataset for a new trigger: assassinations. We describe the process of deciding which ML tool to implement within the framework and subsequently leverage our robust ML framework along with a combination of additional NLP tools, such as NER and KGs, to create the SPE dataset. By mitigating the drawbacks and uniting the strengths of both machine-based and human-centric approaches we create the most comprehensive (targeting all known assassination events), consolidated (a single dataset solely focused on assassination events), and cohesive (easily filterable and readable) assassinations dataset to provide temporal data for understanding political violence as well as training data for further socio-political research [1].

2 Related Work

2.1 Existing Assassinations Datasets

To date, there is no dataset created with the sole intent of targeting all global assassinations of leadership figures. There are pre-existing datasets that either include assassination events as a small portion of the data entries or small scale case studies focusing on specific assassination events in a given country. Nevertheless, there are two previously existing dataset that we explored for assassination events: (1) the Archigos dataset (Goemans et al., 2009) and (2) the Global Terrorism Database (GTD) (LaFree and Dugan, 2007).

Created in 2009, Archigos serves primarily as a data set of political leaders in 188 countries from 1875 to 2015 and has 1,287 entries in its latest version (4.1). Each entry contains the political leader's name, age, gender, term start date and end date, and fate a year after leaving office. The GTD is more comprehensive, as it contains over 200,000 terrorism event entries from 171 countries in the years 1970 to 2019. This data was retrieved from approximately four million global news articles. Each entry contains the date, location, weapons used, target, number of casualties, and group or individuals responsible; but unfortunately, often includes a position description (i.e. mayor) as opposed to the name for the assassination target ('target1' column in dataset).

2.2 Existing Tools for SPE Extraction

Although no comprehensive assassination dataset is available, building robust SPE databases is not a new task of interest and the tools used to create these databases have varied. Many of these more established databases, as well as some newer databases, are manually coded by humans (Raleigh et al., 2010; Gleditsch et al., 2002; Kriesi et al., 2020). These human in the loop projects require full-time permanent employees and extensive support and funding due to the large amount of data to code. For example, Gleditsch et al. (2002) staff processes nearly 50,000 news items and other reports yearly. To mitigate these challenges, many SPE projects have relied on automated event coders like KEDS (Schrodt et al., 1994) or PETRARCH (Schrodt et al., 2014) to record political events (Leetaru and Schrodt, 2013; Halterman et al., 2017). Although these tools provide increased automation, they produce further challenges, such as bias due to human-curated dictionaries, the inability for replication, and issues with aggregating multiple reports into a single event (Rød and Weidmann, 2013). Therefore, many SPE projects have shifted focus to new ML frameworks.

Some projects leverage a hybrid approach of human coding and ML such as Nardulli et al. (2015) to curate a Social, Political and Economic Event Database and Pavlick et al. (2016) to curate a gun violence database. Other projects focus strictly on ML, such as using BERT-based models to extract protest events (Caselli et al., 2021; Celik et al., 2021; Hanna, 2017; Büyüköz et al., 2020). Researchers have also incorporated NER and pre-existing databases along with the ML tools to perform distant supervision such as Reschke et al. (2014) to create a plane crashes database and Keith et al. (2017) to create a police killings dataset. Finally, KGs have been leveraged by Rudnik et al. (2019) to create an event search engine and other researchers have began combining ML and KG for engineering datasets (Guo et al., 2020; Subasic et al., 2019) but to our knowledge there are no examples of this specific combination in the SPE domain.

These hybrid ML methodologies either rely on the availability of many trained human readers, large training datasets, or structured and dense existing datasets for distant supervision. Our project, on the other hand, is focused on minimizing human labor, leveraging a small training dataset, and

[1] Upon the completion of the blind review process our dataset will be released publicly at the conference through our university curation system.

building off incomplete datasets and therefore calls for a novel hybrid approach to dataset engineering that leverages a ML framework for small training sets, NER, KGs, and human-centric approaches.

2.3 Choosing a ML Tool

Researches have implemented traditional ML tools, such as SVMs, K-nearest neighbor, Decision Trees, and Naive Bayes for text classification. From these tools, we selected SVMs as the baseline for our project based on background research that shows that SVMs often outperform other text machine learning tools due to their "simple structure, complete theory, high adaptability, global optimization, short training time, and good generalization performance" (Liu et al., 2010; Gayathri and Marimuthu, 2013; Kwok, 1998; Wright et al., 2013). We also experimented with neural network architecture such as CNNs, RNNs, and LSTMs, but whereas SVMs are equipped to train on smaller datasets (Díaz Rodríguez et al., 2004; Gao and Sun, 2010; Zhang et al., 2008), these models require larger training sets than were available [2].

Newer NLP neural network tools include word embedding tools such as word2vec (Mikolov et al., 2013) and transformers (Vaswani et al., 2017) such as BERT (Devlin et al., 2019). BERT is a transformer based NLP tool that was pre-trained through masked language modeling and next sentence prediction tasks using 3.3 Billion words total with 2.5B from Wikipedia and 0.8B from BooksCorpus (Devlin et al., 2019). The model can be fine-tuned using labeled text for different downstream NLP tasks, such a classification (González-Carvajal and Garrido-Merchán, 2020). Since this is such a powerful and efficient model, there have been countless variants of BERT which can be viewed on the Huggingface library (Wolf et al., 2020). In this study we will focus on 1) BERT-base, a smaller version of the BERT model released by Google, which we will refer to as our 'BERT model' and 2) Longformer, a BERT-based model that aims to handle inputs of longer length by using segment-level recurrence mechanisms to capture information from all the tokens of a document (Beltagy et al., 2020). With Longformer each document can be represented by up to 4,096 tokens, as opposed to 512 for BERT, so we hoped leveraging Longformer would rescue information from our long text inputs that was potentially lost when implementing BERT.

There have been several studies comparing SVMs and pre-trained BERT models for SPE extraction. Olsson et al. (2020); Büyüköz et al. (2020) find that BERT-based models outperform tradition ML algorithms while Piskorski and Jacquet (2020) finds that tf-idf-weighted character n-gram SVM models outperform BERT-based models. It is important to note that Olsson et al. (2020); Büyüköz et al. (2020) and other SPE projects that focus solely on pre-trained models, such as Caselli et al. (2021); Celik et al. (2021), have significantly larger training sets available than our project.

2.4 Wikidata

Knowledge graphs leverage graph structure to represent data where edges capture the relations between entities within the data which allows researchers to extract knowledge, such as events, from the structure (Hogan et al., 2021). The KG that we leverage is Wikidata, which contains over 96 million data items that are expressed through property-value pairs, so each item can have many different properties associated with it. Vrandečić and Krötzsch (2014) discuss some of the applications of Wikidata, including browsing and querying the data it contains. Wikidata also provides an interface for access as a directed labeled graph using the RDF data model and SPARQL query language [3]. Some of the most cited issues with large knowledge graphs like Wikidata include "maintaining their coverage, correctness, and freshness" (Hur et al., 2021), challenges that will be mitigated through our hybrid engineering approach.

3 A Robust ML Framework

3.1 Dataset for Refining Framework: Coups

In order to refine a robust machine learning framework and highlight challenges along the way we chose to focus on one trigger, namely coups. We chose this trigger because it had the highest overlap in classification by humans at the time with an intercoder reliability score of 87.50% agreement. The coup data consists of the English-translated text of news articles retrieved via LexisNexis queries based on several search parameters: a date filter from 1989-2017, a source filter for our list of 20 sources, and keywords, such as 'coup' and 'over-

[2] Models only reached scores of 66.4% (CNN), 76.3% (RNN), and 61.8% (LSTM) with validation losses stagnating above 50 percent or rising dramatically during training for coups

[3] https://query.wikidata.org

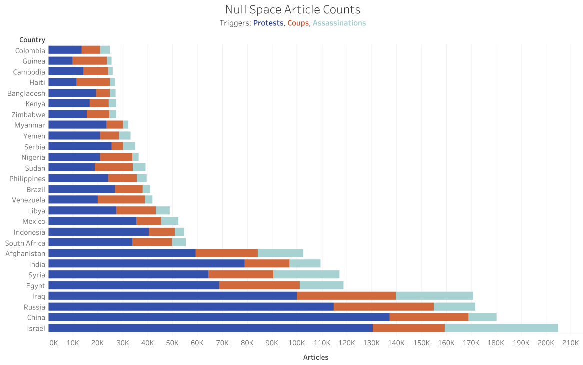


Figure 1: Article counts for countries that comprise >1% of the total articles pulled down across each trigger.

throw', based on trigger definition[4]. The corpus that we hope to classify contains 647,989 unclassified articles. This large magnitude of queried articles (Fig 1), even for a trigger with high intercoder scores and simple keywords, highlights the importance of defining the event of focus with amazing clarity to enable a precise query. In addition to the unclassified dataset, we used a training set consisting of 551 articles (117 positive and 434 negative) retrieved in the same manner and labeled by a team of researchers trained to identify articles that qualify as a coup event.

3.2 Event Coding with PETRARCH

In the beginning stages of the project we leveraged the PETRARCH (Schrodt et al., 2014) event coding software to search for specific key word associations that are defined within a custom definition file fed to the PETRARCH software. These dictionary files are unique to a given trigger and follow the CAMEO standard for political event extraction (Gerner et al., 2002). We employed the trigger coding definitions from Burley et al. (2020) which included the specific key words for coups. During the dictionary creation process, we found that creating new dictionaries for each refinement of a search is labor intensive and risks added bias. This motivated us to shift towards newer machine learning methods to develop an inference engine to gather articles that fit our trigger definitions.

3.3 Classification with SVMs

Our overall ML framework (Fig 2) is split into two phases: the development phase and the production phase. The development phase involves training, testing, and iteratively tuning the machine learning algorithm which allows each model to 'learn' the patterns in the data that separate an instance of a potential trigger versus a non-trigger. Once a model is sufficiently optimized, we classify our larger, unlabeled data set in the production phase.

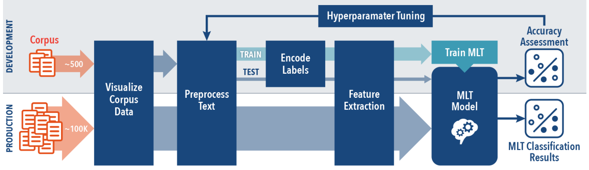


Figure 2: ML framework for article classification.

Our SVM workflow script was initially modeled off of a concise text classification example written by Gungit Bedi (Bedi, 2019). The workflow begins with robust visualizations of the data, as these can aid in understanding the textual relationships from which the machine learning algorithms will produce insights. Next, the text is preprocessed: blank rows removed, text lowercased, stopwords removed, and text tokenized and lemmatized. After these steps, and once the labels are encoded, the processed text is transformed into a numerical vector that can be understood and utilized in the SVM algorithm. The tf-idf vectorizer builds a vocabulary by transforming the articles into a tf-idf-weighted document-term sparse matrix of size (n_articles, m_features). Within the matrix, a higher tf-idf value denotes a stronger relationship between a term and the document in which it appears (Lilleberg et al., 2015). Finally, both the encoded labels and text vectors are inputted into the SVM model where the model trains and learns from the data. After finding optimal training hyperparameters via a grid-search, we ultimately set the training percentage = 80%, C-value = 1, and kernel type = linear.

3.4 Classification with BERT and Longformer

The framework for training our BERT and Longformer is the same as Figure (2), making our pipeline understandable and reproducible. The scripts for BERT and Longformer are based on a tutorial provided by Venelin Valkov (Valkov, 2020). We decided to train the Longformer in addition to BERT due to the high percentage of articles over the 512 token limit for BERT (Fig 3).

The BERT-based preprocessing begins similarly to our SVM as the data is imported and blank rows are removed. Conveniently, the Hugging-

[4] Please contact authors for robust trigger definitions and associated keyword

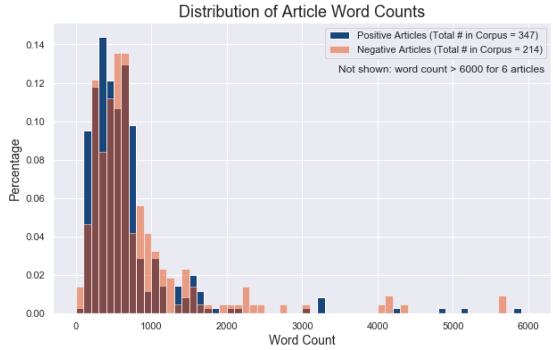


Figure 3: Distribution of training word counts by class.

Face Library provides tokenizers for each model which pre-process the text. Under the hood, these tokenizers lowercases all words and decomposes the input into individual words. More precisely, the BERT tokenizer decomposes inputs into word-pieces (Wordpiece tokenization) while the Longformer tokenizer decomposes words via byte-pair encoding. Since the BERT model uses the original text data to gain understanding of long-term dependencies between words, vectorizing with tf-idf is unnecessary. Rather, the tokenizer simply transforms the tokens to their corresponding integer ids. There are several special tokens added to each input, such as [PAD] which is added to the end of inputs to make each entry the same length, but all other tokens are integer IDs given to each word based on the WordPiece embeddings vocabulary. These input IDs, along with an attention mask are passed to each of the BERT-based models.

The training of the BERT model is more abstract than the SVM. The BERT and Longformer pre-trained weights were downloaded from the pretrained models named ‘bert-base-uncased’ and ‘allenai/longformer-base-4096’ on the Hugging-Face library, respectively. Then a dropout layer and a final linear layer for classification was added to each of these models. We closely followed the original BERT hyperparameters in our script, specifically, sparse categorical cross-entropy as the loss function, ADAM for the optimization algorithm, a batch size of 6, a learning rate of 2e-5, and 50 epochs. A maximum token length of 512 was used for the BERT model and a length of 1,250 was used for the Longformer model (based on counts in Fig 3) which reduces long inputs down to this maximum length for each model and leaves out remaining tokens.

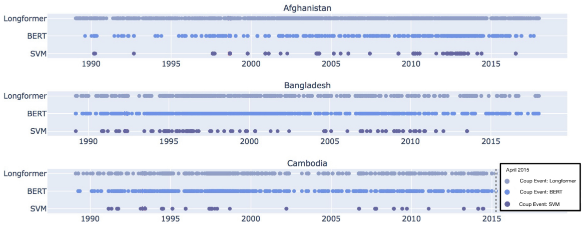


Figure 4: Timeline of articles describing coup events for a subset of countries as classified by each model.

Table 1: Acccuracy Comparison of SVM, BERT, and Longformer Models

	SVM	BERT	Longformer
Number of Positive Coups	28,552	74,871	73,580
Accuracy Score	96.39	96.34	91.67
Precision Score	98.0	95.0	92.0
Human Validation Score	78.93	78.05	77.56

3.5 Comparative Analysis of Models

We used the trained SVM kernels, BERT, and Longformer models to classify the 647,989 unlabeled coup articles and performed a comparative analysis of the results. We created Fig 4 to visualize the classified data and quickly identify differences in how each of the model classify different articles.

Number of Articles Classified as a Coup Event Figure 4 highlights the issue that all models seem to over-specify articles as positive coups (a high false positive rate), shown by the deceiving appearance of constant coup events occurring in each country across 1989-2017. Therefore, we record the total number of articles classified as a coup event (Fig 1, row 1). Evidently, the SVM outperformed BERT and Longformer in terms of refraining from over-specifying articles as coups.

Accuracy Score on Test Data - We compared the predicted labels on the test data to their correct labels (Fig 1, row 2). These scores were extremely promising given our small training dataset.

Precision on Test Data - We produced confusion matrices and classification reports which output precision, recall, and F1 scores. Precision was the most important metric for our project due to the problem of false positives and preference for Type II errors over Type I errors. Maximizing precision minimizes false positive errors. The precision of each model are shown in row 3 of Table 1.

Accuracy Score on Subset of Human Validation Data - A subset of the classification results were validated/coded by the political science researchers. There were 622 articles in this subset, 15 labeled “yes” by the human coders and 607 la-

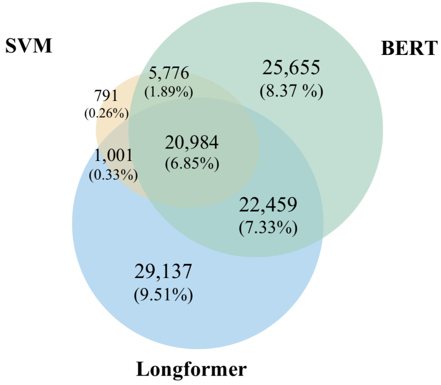


Figure 5: Overlap in model predictions of positively-classified coup articles

beled “no.” We compared these labels to the labels that each model gave to these same 622 articles. These percentages are given in row 3 of Table 1.

Similarity Percentage Between Models In addition to statistical accuracies, it is also useful to analyze the similarities between our 3 models. Specifically, we focused on reporting the overlap of the positive coup articles as shown in Figure 5. We found a 93.72% overlap between SVM and BERT, 59.04% between BERT and Longformer, and 77% between SVM and Longformer. We also found that the "yes" articles could be further decreased from 28,552 to 20,984 articles by taking the overlap of SVM, BERT, and Longformer results where all agree on a positive classification (as opposed to focusing on the SVM classified coup events.)

Resource Restraints The SVM model showed no time or resource restraints. The BERT-based models, on the other hand, took 15 times longer to train than the SVM, and required a GPU for training. Additionally, the batch sizes for both BERT-based models could not exceed the size of 6 due to memory constraints.

Interpretability The SVM proved more interpretable than the BERT-based algorithms. We were able to visualize the most significant tokens for classification as measured by tf-idf scores (Fig 6). We also used a dimensionality reduction algorithm, UMAP (McInnes et al., 2020) to reduce each tf-idf document vector to 2-dimensional vectors and plot these vectors. In the resulting plot the ’yes’ and ’no’, or coup and non-coup, articles are roughly clustered together (Fig 7). The line added to the figure to separate these two clusters is a hypothetical representation of the SVM. These types of tangible representations are not as readily available for the BERT-based models due to their complexity and

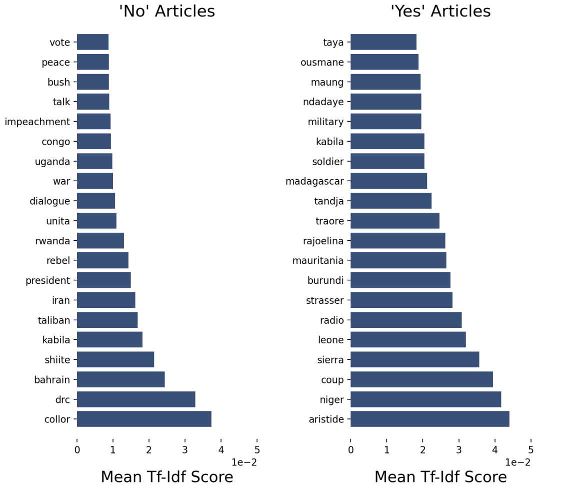


Figure 6: The most significant tokens towards classification of the training set, measured by tf-idf score.

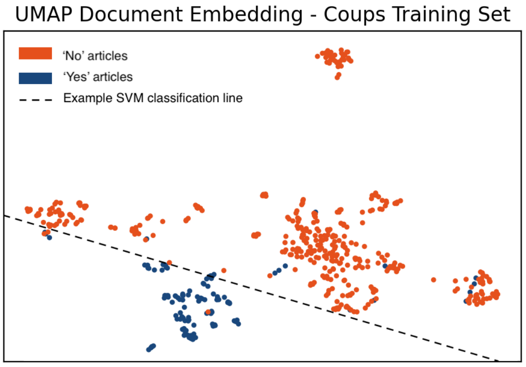


Figure 7: 2D projection of the training set documents with an example SVM classification line.

the pre-trained aspects.

4 Hybrid Knowledge Engineering to Create a SPE Dataset: Assassinations

After refining our ML framework, our next step was to implement the framework on one of SPE of interest in order to create the desired dataset. We switched to assassination events to create our SPE dataset because we saw a lack of assassination datasets in literature (Section 2.1) and assassinations are the most clearly defined trigger[5] for the triggers laid out in (Burley et al., 2020) with one keyword (’assassination’). We leveraged our flushed out machine learning pipeline, along with existing assassination datasets, KGs, and NER to enhance our new assassinations dataset (Fig 8).

[5] Please contact authors for robust trigger definitions and associated keyword

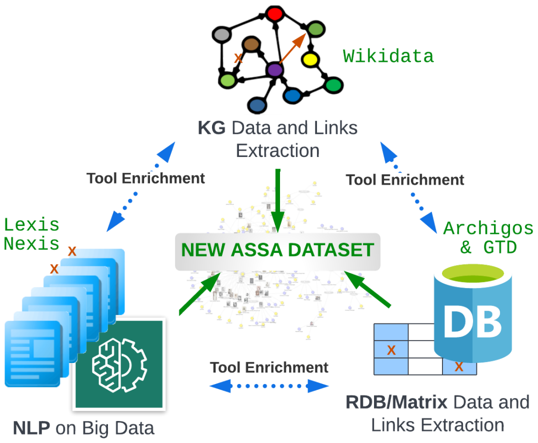


Figure 8: Methodology for linking disparate datasets to build a robust assassinations dataset.

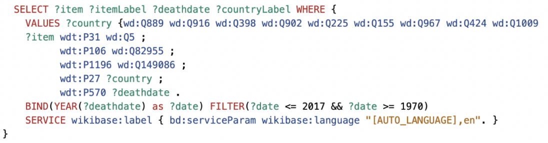
4.1 Existing Assassination Datasets

The Archigos and GTD datasets were the initial contribution to our new assassination dataset. Of the 1,287 entries in the Archigos dataset, 22 of them had “irregular” exits from political office and a post tenure fate marked as “death” with their death year also being the same as their final year of office. It is important to note that natural deaths are marked as regular exits from office, meaning that these irregular exits are actually assassinations.

The GTD contains 6,064 assassination events over the same period of time but includes far more assassinations than simply those of well-known political leaders. The three largest categories for assassinated individuals includes government officials, private citizens, and police. Overall, the GTD contains 6,064 assassinations where 4,442 are successful (target is killed) and 1,622 are unsuccessful (target is not killed).

4.2 Linking in Wikidata

Neither existing database was comprehensive in nature, namely Archigos contained very few assassinations and GTD did not always contain names of the assassinated. We therefore turned to Wikidata to create a stronger baseline for our dataset. For initial exploration of Wikidata, we queried for assassination events, political murders, and deliberate murders using the SPARQL interface. Filters were constructed for the dates ranges and countries of interest, generating 77 results, of which only 55 had a victim associated with them in Wikidata.



```
SELECT ?item ?itemLabel ?deathdate ?countryLabel WHERE {
  VALUES ?country {wd:Q889 wd:Q916 wd:Q398 wd:Q902 wd:Q225 wd:Q155 wd:Q967 wd:Q424 wd:Q1009
  ?item wdt:P31 wd:Q5 ;
        wdt:P106 wd:Q82955 ;
        wdt:P1196 wd:Q149086 ;
        wdt:P27 ?country ;
        wdt:P570 ?deathdate .
  BIND(YEAR(?deathdate) as ?date) FILTER(?date <= 2017 && ?date >= 1970)
  SERVICE wikibase:label { bd:serviceParam wikibase:language "[AUTO_LANGUAGE],en". }
}
```

Figure 9: SPARQL Query to retrieve Wikidata entries.

After querying for events, we queried for victims. We queried for 3 different properties shown in Fig 9: (1) Instance of human (Q5), (2) Date of death (P570) between 1970 and 2017, (3) Manner of death of homicide (Q149086). This resulted in 4,765 individuals. Politician was the most frequent occupation, with 736 individuals, followed by journalist, but there was a large decrease in the frequency of subsequent occupations. We ultimately decided to move forward with just the politician and journalist entries, which gave us 953 victims. Note that these were successful assassinations as the Wikidata methodology does not allow for querying attempted assassinations.

Once we recorded the individual Wikidata “Q” identifiers for the assassination victims, we retrieved the data about each victim using the *qwikidata*[6] library that populates a python dictionary for each Wikidata entity. This allowed us to filter for 5 attributes about the victim: (1) Name, (2) Death Date, (3) Occupation (i.e. politician), (4) Position Held (i.e. Prime Minister of Israel), and (5) Country of Citizenship. Once these Wikidata identifiers were retrieved, we again utilized *qwikidata* to get the Wikidata label strings associated with these entities to populate our dataset.

4.3 Leveraging our ML Framework

To complete our dataset we implemented our ML framework to identify and record all assassination events found in our assassination news data which was pulled with the same LexisNexis query as coups but used assassination keywords. The unclassified corpus consisted of 169,637 unique entries and our training set consisted of 165 humanly labeled articles (76 positive and 89 negative). We trained an SVM, BERT, and Longformer models with our framework since it was necessary to evaluate all models before choosing one or a combination of the models. Both BERT and Longformer performed poorly, with accuracy scores of 64% and 44%, and showed extreme cases of overfitting. The SVM reached an accuracy score of 72.67% which was sufficient considering the human readers only reached 75% in intercoder reliability checks. These

[6] qwikidata: https://qwikidata.readthedocs.io/en/stable/index.html

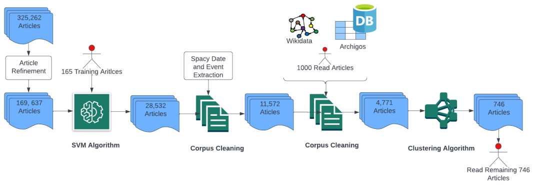


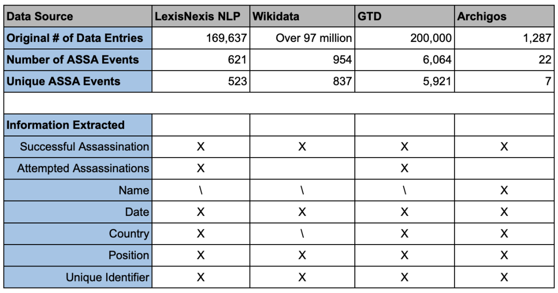
Figure 10: Pipeline for reducing the number of articles read by human readers.

results, along with the high accuracy and precision, smaller number of 'yes' articles, lower resource restraints, and better interpretability shown in Section 3.5, resulted in the use of SVMs to classify the assassination articles.

The trained SVM classified 28,532 articles as assassination events. Similar to Section 3.5, the large magnitude of positively classified assassination articles was a limitation to our ML methodology. So, although ML was leveraged to reduce the number of human read articles, we were still left with nearly 30,000 articles to read through. To rectify this, NER and clustering algorithms were used so reduce the number of human read articles without the need for a larger training dataset (Fig 10).

We first explored SpaCy to refine our assassination event extraction by uploading a pre-trained English pipeline (Honnibal and Montani, 2017) and extracting all names and dates from each positively classified article. This did not assist us in directly identifying assassination events due to the length, complexity, and quantity of names in each article, but during this process, we pinpointed 3 ways to further clean the positively labeled articles: (1) removed articles with text extraction errors (articles with less than 25 words), (2) removed articles with no extracted names, and (3) removed any articles that were nearly duplicates of another by dropping articles that were published within 1 week of another article that had the same subset of extracted names. After this, 11,572 articles remained.

Next, a team of political scientists read 1,000 articles from our original positively labeled articles. The readers classified these articles and recorded all identified assassination events. Based on these events, Wikidata, and Archigos, we removed all articles that contained the person's name of already recorded events. This produced a corpus of 4,771 articles. Next, we clustered articles based on year published and country mentioned in the article and randomly selected one article from each cluster since many articles from a country published in the same year reference the same assassination. Readers read through the remaining 746 articles.



Data Source	LexisNexis NLP	Wikidata	GTD	Archigos
Original # of Data Entries	169,637	Over 97 million	200,000	1,287
Number of ASSA Events	621	954	6,064	22
Unique ASSA Events	523	837	5,921	7
Information Extracted				
Successful Assassination	X	X	X	X
Attempted Assassinations	X		X	
Name	\	\	\	X
Date	X	X	X	X
Country	X	\	X	X
Position	X	X	X	X
Unique Identifier	X	X	X	X

Figure 11: Dataset Summary (For each tool, a given information category was extracted for either all (X), the majority (\), or none (blank) of the extracted events)

5 Results: The Assassination Dataset

By uniting the strengths of each tool within our hybrid approach we created an assassination dataset with 7,457 assassination events. For each entry we collected available information on Name, Date, Country, Position, and Success status (successful vs. attempted) of each assassination event along with the unique identifiers from the source(s) it was identified from. Figure 11 highlights the unique information and number of assassination events contributed by each tools. This shows that despite each method's limitations, ambiguous event definitions for humans, incomplete datasets, missing Wikidata properties, and small training datasets for ML, it is evident that each tool benefited the dataset. Existing databases provided a starting point for our dataset, Wikidata enhanced our repository, and the ML pipeline allows us to extract assassination events from 169,637 articles with only a 165 article training set.

6 Conclusion & Future Work

We have contributed to ongoing SPE research by providing a robust ML framework for small training datasets, performing a comparative analysis of ML tools, presenting a novel hybrid knowledge engineering approach to curate a dataset, and releasing our comprehensive, consolidated, and cohesive assassination dataset which will provide temporal data for understanding political violence as well as training data for further socio-political research. Although our framework and hybrid knowledge engineering approach will not perfectly transfer for every SPE dataset curation task, our focus on understandability visualizations, replicable frameworks, and explanation of challenges will allow future researches to incorporate our work for a variety of SPE extraction tasks. In future work, we hope to apply the knowledge engineering approach, encompassing our ML framework, to the remaining triggers of interest while continuing to improve and automate our ML framework.

7 Acknowledgments

The authors would like to thank Timothy Burley, Sophia Henn, Steven Broussard, Ian Havenaar, Angela Chessler, Dr. Patrick Regan, and Dr. Charles Vardeman for their contributions.

References

Gunjit Bedi. 2019. Simple guide to text classification(nlp) using svm and naive bayes with python. Medium.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer.

Timothy Burley, Lorissa Humble, Charles Sleeper, Abigail Sticha, Angela Chesler, Patrick Regan, Ernesto Verdeja, and Paul Brenner. 2020. Nlp workflows for computational social science: Understanding triggers of state-led mass killings. In *Practice and Experience in Advanced Research Computing*, pages 152–159.

Berfu Büyüköz, Ali Hürriyetoğlu, and Arzucan Özgür. 2020. Analyzing ELMo and DistilBERT on socio-political news classification. In *Proceedings of the Workshop on Automated Extraction of Socio-political Events from News 2020*, pages 9–18, Marseille, France. European Language Resources Association (ELRA).

Tommaso Caselli, Osman Mutlu, Angelo Basile, and Ali Hürriyetoğlu. 2021. Protest-er: Retraining bert for protest event extraction. In *Proceedings of the 4th Workshop on Challenges and Applications of Automated Extraction of Socio-political Events from Text (CASE 2021)*, pages 12–19.

Furkan Celik, Tuğberk Dalkılıç, Fatih Beyhan, and Reyyan Yeniterzi. 2021. Su-nlp at case 2021 task 1: Protest news detection for english. In *Proceedings of the 4th Workshop on Challenges and Applications of Automated Extraction of Socio-political Events from Text (CASE 2021)*, pages 131–137.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Susana Irene Díaz Rodríguez, José Ranilla Pastor, Elena Montañés Roces, Javier Fernández, and Elías Fernández-Combarro Álvarez. 2004. Improving performance of text categorization by combining filtering and support vector machines. *Journal of the American Society for Information Science and Technology, 55 (7)*.

Ya Gao and Shiliang Sun. 2010. An empirical evaluation of linear and nonlinear kernels for text classification using support vector machines. In *2010 seventh international conference on fuzzy systems and knowledge discovery*, volume 4, pages 1502–1505. IEEE.

K Gayathri and A Marimuthu. 2013. Text document pre-processing with the knn for classification using the svm. In *2013 7th International Conference on Intelligent Systems and Control (ISCO)*, pages 453–457. IEEE.

Deborah J Gerner, Philip A Schrodt, Omur Yilmaz, and Rajaa Abu-Jabr. 2002. The creation of cameo (conflict and mediation event observations): An event data framework for a post cold war world. In *annual meeting of the American Political Science Association*, volume 29.

Nils Petter Gleditsch, Peter Wallensteen, Mikael Eriksson, Margareta Sollenberg, and Håvard Strand. 2002. Armed conflict 1946-2001: A new dataset. *Journal of peace research*, 39(5):615–637.

Henk E Goemans, Kristian Skrede Gleditsch, and Giacomo Chiozza. 2009. Introducing archigos: A dataset of political leaders. *Journal of Peace research*, 46(2):269–283.

Santiago González-Carvajal and Eduardo C Garrido-Merchán. 2020. Comparing bert against traditional machine learning text classification. *arXiv preprint arXiv:2005.13012*.

Kaihao Guo, Tianpei Jiang, and Haipeng Zhang. 2020. Knowledge graph enhanced event extraction in financial documents. In *2020 IEEE International Conference on Big Data (Big Data)*, pages 1322–1329. IEEE.

Andrew Halterman, Jill Irvine, Manar Landis, Phanindra Jalla, Yan Liang, Christan Grant, and Mohiuddin Solaimani. 2017. Adaptive scalable pipelines for political event data generation. In *2017 IEEE International Conference on Big Data (Big Data)*, pages 2879–2883. IEEE.

Alex Hanna. 2017. Mpeds: Automating the generation of protest event data.

Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia d'Amato, Gerard De Melo, Claudio Gutierrez, Sabrina Kirrane, José Emilio Labra Gayo, Roberto Navigli, Sebastian Neumaier, et al. 2021. Knowledge graphs. *ACM Computing Surveys (CSUR)*, 54(4):1–37.

Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear.

Ali Hur, Naeem Janjua, and Mohiuddin Ahmed. 2021. A survey on state-of-the-art techniques for knowledge graphs construction and challenges ahead. In *2021 IEEE Fourth International Conference on Artificial Intelligence and Knowledge Engineering (AIKE)*, pages 99–103. IEEE.

Katherine A Keith, Abram Handler, Michael Pinkham, Cara Magliozzi, Joshua McDuffie, and Brendan O'Connor. 2017. Identifying civilians killed by police with distantly supervised entity-event extraction. *arXiv preprint arXiv:1707.07086*.

Hanspeter Kriesi, Edgar Grande, Swen Hutter, Argyrios Altiparmakis, Endre Borbáth, S Bornschier, B Bremer, M Dolezal, T Frey, T Gessler, et al. 2020. Poldem-national election campaign dataset.

James Tin-Yau Kwok. 1998. Automated text categorization using support vector machine. In *In Proceedings of the International Conference on Neural Information Processing (ICONIP*. Citeseer.

Gary LaFree and Laura Dugan. 2007. Introducing the global terrorism database. *Terrorism and political violence*, 19(2):181–204.

Kalev Leetaru and Philip A Schrodt. 2013. Gdelt: Global data on events, location, and tone, 1979–2012. In *ISA annual convention*, volume 2, pages 1–49. Citeseer.

Joseph Lilleberg, Yun Zhu, and Yanqing Zhang. 2015. Support vector machines and word2vec for text classification with semantic features. In *2015 IEEE 14th International Conference on Cognitive Informatics & Cognitive Computing (ICCI* CC)*, pages 136–140. IEEE.

Zhijie Liu, Xueqiang Lv, Kun Liu, and Shuicai Shi. 2010. Study on svm compared with the other text classification methods. In *2010 Second international workshop on education technology and computer science*, volume 1, pages 219–222. IEEE.

Leland McInnes, John Healy, and James Melville. 2020. Umap: Uniform manifold approximation and projection for dimension reduction.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality.

Peter F Nardulli, Scott L Althaus, and Matthew Hayes. 2015. A progressive supervised-learning approach to generating rich civil strife data. *Sociological methodology*, 45(1):148–183.

Fredrik Olsson, Magnus Sahlgren, Fehmi Ben Abdesslem, Ariel Ekgren, and Kristine Eck. 2020. Text categorization for conflict event annotation. In *Proceedings of the Workshop on Automated Extraction of Socio-political Events from News 2020*, pages 19–25.

Ellie Pavlick, Heng Ji, Xiaoman Pan, and Chris Callison-Burch. 2016. The gun violence database: A new task and data set for nlp. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1018–1024.

Jakub Piskorski and Guillaume Jacquet. 2020. Tf-idf character n-grams versus word embedding-based models for fine-grained event classification: a preliminary study. In *Proceedings of the Workshop on Automated Extraction of Socio-political Events from News 2020*, pages 26–34.

Clionadh Raleigh, Andrew Linke, Håvard Hegre, and Joakim Karlsen. 2010. Introducing acled: an armed conflict location and event dataset: special data feature. *Journal of peace research*, 47(5):651–660.

Nitin Ramrakhiyani, Swapnil Hingmire, Sangameshwar Patil, Alok Kumar, and Girish Palshikar. 2021. Extracting events from industrial incident reports. In *Proceedings of the 4th Workshop on Challenges and Applications of Automated Extraction of Socio-political Events from Text (CASE 2021)*, pages 58–67.

Kevin Reschke, Martin Jankowiak, Mihai Surdeanu, Christopher D Manning, and Dan Jurafsky. 2014. Event extraction using distant supervision. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 4527–4531.

Espen Geelmuyden Rød and Nils B Weidmann. 2013. Protesting dictatorship: The mass mobilization in autocracies database. Technical report, Citeseer.

Charlotte Rudnik, Thibault Ehrhart, Olivier Ferret, Denis Teyssou, Raphaël Troncy, and Xavier Tannier. 2019. Searching news articles using an event knowledge graph leveraged by wikidata. In *Companion proceedings of the 2019 world wide web conference*, pages 1232–1239.

Philip A Schrodt, John Beieler, and Muhammed Idris. 2014. Three'sa charm?: Open event data coding with el: Diablo, petrarch, and the open event data alliance. In *ISA Annual Convention*. Citeseer.

Philip A Schrodt, Shannon G Davis, and Judith L Weddle. 1994. Political science: Keds—a program for the machine coding of event data. *Social Science Computer Review*, 12(4):561–587.

Pero Subasic, Hongfeng Yin, and Xiao Lin. 2019. Building knowledge base through deep learning relation extraction and wikidata. In *AAAI Spring Symposium: Combining Machine Learning with Knowledge Engineering*.

Venelin Valkov. 2020. Text classification | sentiment analysis with bert using huggingface, pytorch and python tutorial. YouTube.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need.

Ernesto Verdeja. 2016. Predicting genocide and mass atrocities. *Genocide Studies and Prevention: An International Journal*, 9(3):5.

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Adam Wright, Allison B McCoy, Stanislav Henkin, Abhivyakti Kale, and Dean F Sittig. 2013. Use of a support vector machine for categorizing free-text notes: assessment of accuracy across two institutions. *Journal of the American Medical Informatics Association*, 20(5):887–890.

Wen Zhang, Taketoshi Yoshida, and Xijin Tang. 2008. Text classification based on multi-word with support vector machine. *Knowledge-Based Systems*, 21(8):879–886.