2019 IEEE 5th Intl Conference on Big Data Security on Cloud (BigDataSecurity), IEEE Intl Conference on High Performance and Smart Computing (HPSC), and IEEE Intl Conference on Intelligent Data and Security (IDS)

Automated Verbal-Pattern Extraction from Political News Articles using CAMEO Event Coding Ontology.

Erick Skorupa Parolin*, Sayeed Salam*, Latifur Khan*, Patrick T. Brandt†, Jennifer Holmes†
Department of Computer Science, School of Economic, Political and Policy Sciences†*
The University of Texas at Dallas
Richardson, Texas
{erick.skorupaparolin, sxs149331, lkhan, pbrandt, jholmes}@utdallas.edu

***Abstract*—Structured metadata extraction from raw-text in different domains is gaining strong attention from the research communities and offering good applicable scenarios. In Political Science, these metadata play a significant role in studying and predicting intra- and inter-state relationships and often follows a certain structural representation. Our work exploits CAMEO (Conflict and Mediation Event Observations) ontology to extract events from political news articles. This event will be represented by metadata such as who-did-what-to-whom format. We face a number of challenges in this extraction process due to the incompleteness of CAMEO dictionaries. In particular, for some verb patterns appeared in articles we may not find an appropriate match in CAMEO verb dictionary (i.e ”what”). Our goal is to recommend new verb patterns by mining a large number of news articles to enhance the existing CAMEO verb dictionaries. For this, first we apply Association Rule Mining to find frequent verb patterns. Next, to assign categories of these verb patterns, we develop a novel algorithm based on Word-to-Vec model. Finally, we develop a prototype system for automatically recommending a number of useful verb patterns.**

***Index Terms*—CAMEO; Universal Dependencies; Political Event Coding; Association Mining Rules; Word-to-Vec.**

I. Introduction

Structured metadata generation now has profound impact on research areas including social and political science, economics, etc. For example, researchers in political science are focusing on different datasets representing events to study political interaction beteween diffent intra- and inter-state entities. These studies are often backed by organized datasets. [1]–[3]. Structured representation of these events made it easier for desigining software and tools to do analytics and visulaization and thus attracts people from different technical background to the use of the system.

Most of these datasets involves data extraction and gathering phase where structured metadata is captured using annotation procedure. This procedure can be driven by human [2] or be automated by software [3]. Often the automated software relies on well defiend ontologies like Conflict and Mediation Event Observation (CAMEO) and generate events in the form of ”who-did-what-to-whom” format. The w's are indentified using a list of entries in corressponding knowledge-base. The ”who” and ”whom” are identified using CAMEO's list of political actors. The ”what” is identifed using CAMEO's verb-pattern dictionary. Events are identified when all the required information is gathered by the event coder (See Figure 1) and classified to one of the 20 available categories (also known as root codes, refer to Table II).

The information defined under CAMEO ontology is static and new political actor and verb-patterns are not included in the knowledge-base in a regular timely manner. This leads to the failure to capture important events and reduces the coverage of the generated dataset. To overcome this, we need to dynamically update or suggest inclusion of new information (i.e. new actors and verb patterns).

Although previous works related to stream classification and image annotation using ontologies [4]–[26] had shown remarkable results, ontology extension [27], [28] related works in political science domain is still an area to be explored. Solaimani et al. [29] propose a system that can recommend new political actors for inclusion in the CAMEO's actor dictionaries. That work is motivated by the less coverage offered by existing research works [3], [30] as they have very low percentage of the articles generting events (around 7%). Solaimani et al. [29] shown the system to work towards increasing coverage but that alone can not achieve the actual expansion possible. During our initial analysis we found a lot of evidences where the event generation fails due to missing patterns or verbs from the dictionary. This motivated us to address the problem of extending the verb-pattern dictionary of CAMEO ontology.

In this paper, we focus on identifying new verb-patterns which are related to known political actors within a sentence and apply our algorithm to find out most suitable patterns and their corressponding event type with respect to root code in CAMEO. We use association rule mining to identify the (verb,word) pairs happening more frequently. To categorize them to root codes, we observe semantic similarity with listed patterns under different root codes. We use word-to-vec based similarity calculation at this step.

The key contributions in the paper are as follows

- We designed a system for automating the extraction of

978-1-7281-0006-7/19/$31.00 ©2019 IEEE
DOI 10.1109/BigDataSecurity-HPSC-IDS.2019.00056

verbal-patterns from news articles by applying Association Rules Mining approach. These new found patterns are candidates for CAMEO verb dictionary extention.

- We present a novel algorithm for recommending the type of political interaction for new found verb-patterns by applying Word-to-Vec techniques. Assuming our goal is to extend an ontology, only finding new verb patterns without their meaning or type of interaction may not be enough for this purpose.

The remaining part of this paper is organized as follows: Section-II provides idea on key concepts/tools used in the paper. Section-III introduces our proposed approach. Section-IV shows results and output examples. In Section-V we provide concluding remarks and future work.

II. Background

To help the reader better understand the tools and methods used in this paper, we will provide key details of different components and concepts.

Political Event is a structured formation of information consisting of an action, source (acting entity), target (entity being acted upon) and other related information. For example consider the following sentence from a news article -

```
PM Theresa May has struck a last-minute
deal with the EU in a bid to move Brexit
talks on to the next phase.
```

As a structured event, it looks like the following

```
Source - GBRGOV,
Target - IGOEUREEC
Action - 057 (Sign formal agreement)
```

PM Theresa May and EU is coded in standard format used for event generation. The structure used here is mostly known as *who-did-what-to-whom* format of event coding. The coding mechanism is depicted in the Figure 1.

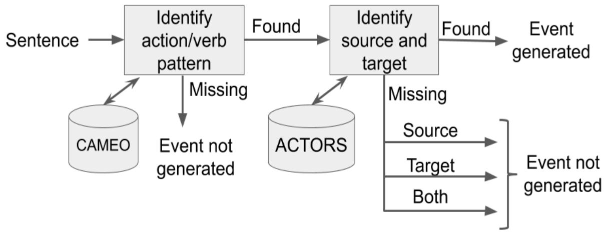

Fig. 1. Basic Mechanism of Automated Coding

Given a sentence, the encoder search for a matching pattern in the CAMEO verb patterns dictionary. A pattern consists of a verb and surrounding keywords. Together they signify a particular course of action and represented with event code. For example, the verb "SET" in the following pattern "OUT VIEWS"

```
SET OUT VIEWS
```

indicates MAKE PUBLIC STATEMENT type of event (event code 010). Upon finding a match in pattern, actor dictionaries are searched for matching entities representing source and target. After finding all the pieces of information, an event is coded by the event coder. If there is missing information, event coder will ignore the event. Table I expresses some statistics computed during our experiments. In summary, we were able to find events for only 17.74% of the processed sentences.

TABLE I
Statistics for coding events.

	#	%
Total Articles	250,000	-
Sentences	4,266,661	100,00%
Sentences with Event	756,921	17.74%

This form of events are also known as Source-Action-Target or SAT format.

Conflict and Mediation Event Observations (CAMEO) is an ontology developed to capture political events and focuses on the following 4 types.

- Verbal Cooperation
- Material Cooperation
- Verbal Conflict
- Material Conflict.

Each of them are distilled further and thus generated 20 root code based categories. It works using a knowledge-base which consists of pattern and actor dictionaries. Pattern dictionary is helpful for identifying political interactions in a given sentence. Actor dictionary is used for searching political actors found around the matched pattern, within a sentence.

Universal Dependency (UD) [31] is an initiative that is developing cross-linguistically consistent tree-bank annotation for many languages, with the goal of facilitating multilingual parser development, cross-lingual learning, and parsing research from a language typology perspective.

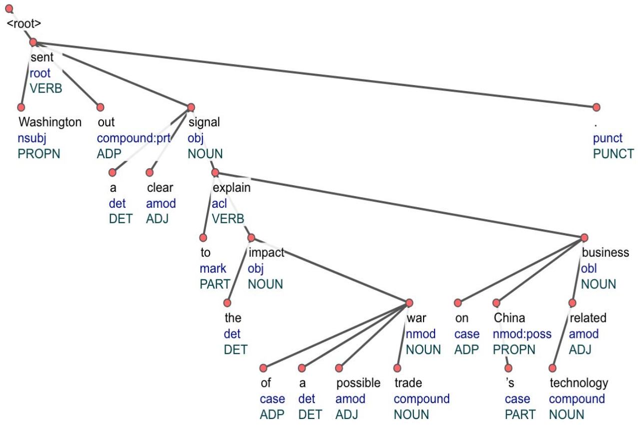

Fig. 2. Universal Dependency parse for Sentence 1

The general goal is to provide a universal collection of categories and guidelines to facilitate consistent annotation of similar constructions across languages, while allowing language-specific extensions when necessary. Figure 2 shows an example of the dependency parsing.

Word-To-Vec or word embeddings [32] is an intersting idea to represent words into numeric vectors such a way that semantic relation between those words can be expressed in-terms of vector operations. For example, the word ”good” and ”fair” is close in meaning, they will have a similar vector and distance between them will be less. This embedding can be derived at the phrase and sentence level

Association Rules Mining is a data mining [33]–[36] technique that consists on identifying frequent patterns, associations, correlations or causal relation among sets of elements usually extracted from analytical data-sets. Specifically in this paper, we apply *apriori* algorithm [37] over sets of words surrounding verbs in input sentences in order to identify candidate verbal-patterns. As a brief example, consider the following sentences

Sentence 1: *Washington* ***sent*** *out a clear* ***signal*** *to explain the impact of a possible trade war on Chinas technology related business.*

Sentence 2: *The King Goodwill Zwelithini offer* ***sends*** *a* ***signal*** *to South Africa’s ruling African National Congress, which intends to redistribute farmland to black property owners.*

Based on the sets of words [Washington, sent, out, signal, explain] and [offer, sends, signal] extracted from both sentences, it is clear that words **send** and **signal** are associated (their occurence in a sentences imply each other’s occurence).

III. Proposed Framework

The flow in Figure 3 depicts each step of the proposed system for generating new verbal-patterns in order to extend the CAMEO event coding ontology.

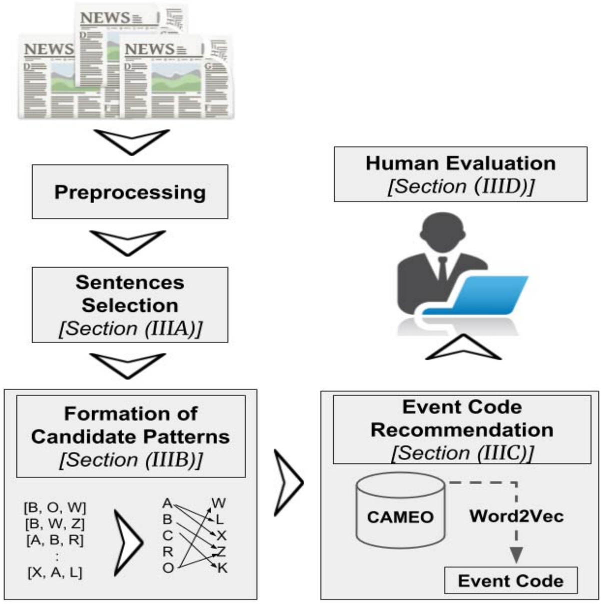
NEWS
NEWS
NEWS
Preprocessing
Sentences Selection [Section (IIIA)]
Formation of Candidate Patterns [Section (IIIB)]
[B, O, W]
[B, W, Z]
[A, B, R]
:
[X, A, L]
A
B
C
R
O
W
L
X
Z
K
Event Code Recommendation [Section (IIIC)]
CAMEO
Word2Vec
Event Code
Human Evaluation [Section (IIID)]

Fig. 3. Automated Verb-Pattern Generator.

We collect news articles from online news websites and in the preprocessing step, we generate universal dependency parse for each of the sentences in those articles. Then we run them through the automated event coder and capture the sentences where source and target of the event are identified but no verb or verb-pattern is found. We collect these sentences and focus on the verb and surrounding words. We consider co-occurance of the verb and words to form the verb-pattern and use association rule minning to identify the candidate pairs. The particular algorithm we use in this step is apriori [37].

Finally, we take advantage of Word-to-Vec [32] technique to compare the obtained candidate verb-patterns against already known CAMEO verb-patterns to derive most appropriate root code class.

The Algorithm 1 introduces the proposed Automated Verbal-Pattern Extraction following those same steps performed in Figure 3.

```
Algorithm 1: Automated Verbal-Pattern Extraction
input  : Set of Articles $articles$, List of patterns in
         CAMEO verb dictionary $cameo$
output : List of Verb Candidate Patterns $candidates$
         with recommended root-codes.
1  $Parsed\_Q \leftarrow \varnothing$ // empty queue
2  $Sets\_Q \leftarrow \varnothing$ // empty queue
3  foreach $article$ in $articles$ do
4      foreach $sentence$ in $article$ do
5          $parsed \leftarrow PREPROCESS(sentence)$
6          $Parsed\_Q.push(parsed)$
7  $selected \leftarrow SENT\_SELECTION(Parsed\_Q)$
8  foreach $parsed\_sentence$ in $selected$ do
9      $set \leftarrow SET\_FROM\_SENT(parsed\_sentence)$
10     $Sets\_Q.push(set)$
11 $rules \leftarrow APRIORI(Sets\_Q)$
12 return $NN\_WORD2VEC(rules, cameo)$
```

Analogously to the flow depicted in Figure 3, we start by preprocessing all sentences (lines 3 - 6) for each news articles given as input, obtaining their correponding dependency trees. Then we select only those parsed sentences we are interested for our analysis through Procedure $SENT\_SELECTION$ (line 7), which will be introduced in section III-A. We use Procedure SET_FROM_SENT to convert the parsed sentences into sets of words (line 8-10) which will be used as Association Mining Rules input. Please see Section III-B for more details about procedure SET_FROM_SENT. These sets of words serve as input for $apriori$ algorithm(line 11). Finally, we take the rules output from $apriori$ as candidate patterns and recommend a root code for each one of them through $NN\_WORD2VEC$ procedure, introduced in Section III-C.

A. Sentences Selection

Recall that the main goal of the proposed framework is to generate specific verbal-patterns which are not found in CAMEO verb-dictionary. Therefore, we must ignore those sentences for which any event was generated and focus on the other sentences which had source and target actors properly identified but no event was generated. In other words, the sentence selection step consists of identifying sentences that belong to political domain but no event was generated just because of the fact that no verbal-pattern was matched in the CAMEO verb dictionary.

The flow shown in Figure 4 summarizes the process of extracting the sentences required for our further analysis. Besides, procedure $SENT\_SELECTION$ presented in Algorithm 2 expresses the pseudo-code version of the sentences selection step.

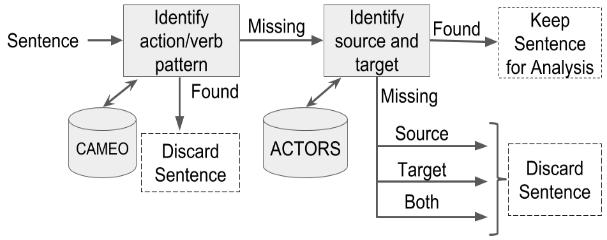

Fig. 4. Modified Mechanism of Automated Coding to extract sentences with no generated event.

Note that the flow for sentences selection step consists on a slight modification of the Basic Mechanism of Automated Coding, illustrated in Figure 1.

Algorithm 2: Procedure $SENT\_SELECTION$

```
input  : Queue $Parsed\_Q$ of parsed sentences
output : Queue $Selected\_Q$ with selected parsed sentences
$Selected\_Q \leftarrow \varnothing$ // empty queue
foreach $parsed$ in $Parsed\_Q$ do
    if not $FOUND\_VERB\_PAT\,(parsed.sentence)$ then
        if $FOUND\_SOURCE\,(parsed.sentence)$ then
            if $FOUND\_TARGET\,(parsed.sentence)$ then
                $Selected\_Q.push(parsed)$
return $Selected\_Q$
```

The procedure $SENT\_SELECTION$ clearly performs similar steps as those in flow depicted in Figure 4. Please note that functions $FOUND\_VERB\_PAT$, $FOUND\_VERB\_SOURCE$ and $FOUND\_VERB\_TARGET$ are simply boolean functions that return true if verb patterns, source and target respectively are matched in CAMEO dictionaries, and false otherwise. These functions are derived from other functionalities available in Universal Dependency PETRARCH [38], which is the mechanism of automated political event coding that was in for our system.

B. Formation of Candidate Patterns

The next task after identifying the sentences with no matched verb-patterns is to transform the parsed sentences into sets of words such that *apriori* algorithm can take them as input to generate candidate patterns based on output rules.

Note that there may be multiple ways of extracting sets of words from sentences. In fact, many combinations of sets can be extracted from the same sentence.

Since we are specifically looking for new verb patterns, the sentences conversion into sets of words is performed based on universal dependency trees. Thus, we can easily identify those words which are dependent or depending on the verbs, increasing the chance of finding a pattern encompassing verbs. In summary, the set of words generated from a given selected sentence contains the verb for which was not found any pattern in CAMEO dictionary and all the adjacent words of this verb in the universal depencendy tree. Algorithm 3 defines the sequence of operations performed in this step.

Algorithm 3: Procedure SET_FROM_SENT

```
input  : A parsed sentence structure $parsed\_sentence$
output : Set of words corresponding to $parsed\_sentence$
$Set\_Q \leftarrow \varnothing$ // empty queue
$Set\_Q.push\,(parsed\_sentence.verb)$
foreach Adjacent word $w$ of $parsed\_sentence.verb$ in $parsed\_sentence.tree$ do
    $Set\_Q.push\,(w)$
return $Set\_Q$
```

Procedure SET_FROM_SENT receives as input an object $parsed\_sentence$ which is composed by the $tree$ (universal dependency tree) corresponding to the sentence and the main $verb$ for which no verb pattern was found CAMEO dictionary. Then, we simply traverse the dependency $tree$ selecting all the words corresponding to adjacent nodes of the $verb$ node in this $tree$. In other words, we push on queue the $verb$ and the words surrounding the $verb$ in the $tree$.

To make this step more clear to the reader, we show in Figure 5 an example illustrating how to capture the set of words corresponding to the sentence below.

Sentence: *Washington* ***sent*** *out a clear* ***signal*** *to explain the impact of a possible trade war on Chinas technology related business.*

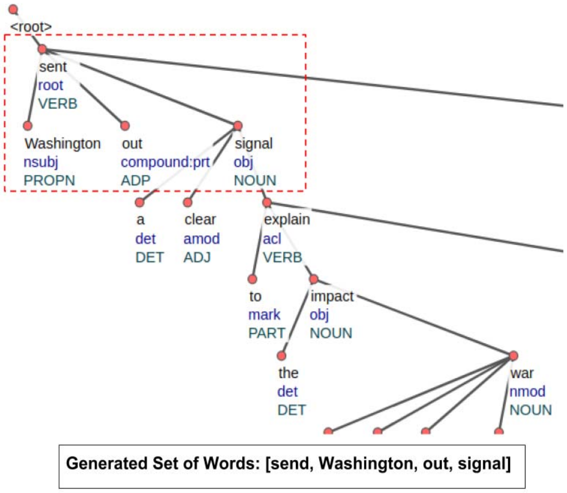

Fig. 5. Capturing Set of Words from Universa Dependency Tree of the Sentences.

We keep analyzing the same sentence aforementioned in Section II in order to evidence how that set of words was obtained. All the words collected for the set are surrounding the verb **sent** in universal dependency tree.

Note that there may be more than one set of words generated from the same sentence. Besides, it is important to mention that an intermediate pre-processing step is run in order to deduplicate words, convert them to lowercase, remove common words and proper nouns, etc.

Finally, after converting all the parsed sentences into sets of words, we apply *apriori* algorithm on them, obtaining as output a association rules, or sets of most frequent co-existing words among the input sets, which we interpret as candidate verb-patterns.

C. Event Code Recommendation

Given the large amount of candidate verb-patterns potentially generated, the task of identifying the type of political interaction corresponding to each candidate would require exhaustive human effort. Therefore, besides identifying the candidate patterns, our proposed system also labels each one of them, recommending a type of interaction based on the CAMEO root-codes [39], which are introduced in Table II.

For this recommendation, we apply word embeddings technique, which allows us to work with vector representation corresponding to words in order to better evaluate semantic relationships between existing and candidate patterns.

The idea behind the task of recommending a root code for candidate patterns is better described by the Procedure $NN\_WORD2VEC$ shown in Algorithm 4. This procedure takes the candidate verb patterns as input and for each one of them, it looks for the k nearest neighbors [40] in the existing CAMEO patterns in order to recommend a root code. The search for these neighbors is performed based on the vector space correspoding to the words in the existing and candidate patterns. As aforementioned, we use Word-to-Vec technique to obtain the vector representation of patterns and the models used for such conversion was chosen based on the experimental results presented in Section IV-C. Besides, the value we use for k was obtained from empirical experiments also shown in Section IV-C.

TABLE II
LIST OF ROOT CODES AND THEIR CORRESPONDING EVENT DESCRIPTIONS.

CAMEO root-codes	Corresponding Interaction Type
01	MAKE PUBLIC STATEMENT
02	APPEAL
03	EXPRESS INTENT TO COOPERATE
04	CONSULT
05	ENGAGE IN DIPLOMATIC COOPERATION
06	ENGAGE IN MATERIAL COOPERATION
07	PROVIDE AID
08	YIELD
09	INVESTIGATE
10	DEMAND
11	DISAPPROVE
12	REJECT
13	THREATEN
14	PROTEST
15	EXHIBIT MILITARY POSTURE
16	REDUCE RELATIONS
17	COERCE
18	ASSAULT
19	FIGHT
20	ENGAGE IN UNCONVENTIONAL MASS VIOLENCE

In Algorithm 4, the outer loop expressed in line 4 goes over all the input rules, which we can read as candidate patterns, and guarantees that all them will be properly labeled by a root code recommendation. Line 6 computes the semantic relationship between the candidate and existing patterns by cosine similarity metric. Note that, although it is not explicitly expressed in this pseudo-code, cosine similarity calculation (performed by function COS) is performed over the vectors corresponding to the words in the CAMEO existing patterns and the candidate pattern. Following, in line 7, we assign the similarity value as key and the root code of the existing pattern involved in the similarity calculation, as value to a $tuple$. Then line 8 pushes this $tuple$ into a Min-Priority Queue NN_Q.

Note that, given similarity is the key of $tuple$, NN_Q will always pop the $tuple$ corresponding to the existing pattern with the lowest similarity among those tuples in the queue. Therefore, lines between 8 and 10 guarantees that the Min-Priority Queue NN_Q always keeps the tuples corresponding to the k nearest neighbors of the candidate which is being analyzed at that time.

Finally, between lines 11 and 15, we identify through $MODE$ function what is the root code that is most frequent among those k nearest existing patterns and recommend it for the candidate that is being analyzed on that iteration. If there is no a unique most frequent root code, we simply select the root code corresponding to the nearest one existing pattern as event code recommendation.

Algorithm 4: Procedure $NN\_WORD2VEC$

```
input   : Set of rules output from APRIORI $rules$,
          List of patterns in CAMEO verb dictionary
          $cameo$, Number $k$ of neighbors to be
          analyzed
output  : List of Verb Candidate Patterns $candidates$
          with recommended root-codes.

$NN\_Q \leftarrow \varnothing$ // Empty Min-Priority Queue
$Root\_Codes\_Q \leftarrow \varnothing$ // Empty Queue
$candidates\_Q \leftarrow \varnothing$ // Empty Queue

foreach $rule$ in $rules$ do
    foreach $existing$ in $cameo$ do
        $sim \leftarrow COS(rule.pattern, existing.pattern)$
        $tuple \leftarrow (sim, existing.code)$
        $NN\_Q.push(tuple)$
        if $LENGTH(NN\_Q) < k$ then
            $lowest\_sim \leftarrow NN\_Q.pop()$

    for $i \leftarrow 1$ to $k$ do
        $nearest \leftarrow NN\_Q.pop()$
        $Root\_Codes\_Q.push(nearest.code)$
    try:
        $root\_code \leftarrow MODE(Root\_Codes\_Q)$
    catch NoUniqueModeFound:
    $candidate \leftarrow (rule.pattern, root\_code)$
    $candidates\_Q.push((candidate))$

return $candidate$
```

D. Human-Annotation

Although the proposed framework recommends event-codes with an acceptable accuracy, it still does not provide enough confidence for adding the output candidates directly to dictionary without a human expert examination. Besides eventual misclassifications, human inspection is also required to evaluate how semantically useful are the generated verb-patterns and evetually review the sentences containing such patterns.

IV. Results and Discussion

A. Environment Setup

All the computational experiments presented in this section were performed in a computer with the following configuration:

- Processor: Intel Core i7-6820HQ, 2.70GHz
- Memory: 16 GB RAM memory
- Operational System: Ubuntu 14.04.5 LTS

The coding task was all performed using Python 3.6.4, $Gensim$ library for Word-to-Vec algorithms and $efficient$-$apriori$ 0.4.5 library for apriori algorithm. Besides, we use Universal PETRARCH [38] as our mechanism of automated coding and UDpipe [41] as universal dependency parser.

B. Data Source

The data used as input for the experiments was collected using URLs from the RSS Feed of around 400 different news agencies spread around the world. After collecting data in regular HTML form, we clean and extract main stories using the tools available in [42]. The a web- scraper [43] program is used for collecting and preprocessing the data used for the following experiments.

Overall, for this experiment, we captured $250,000$ articles. After cleaning the data and filtering out those sentences we are not interested, as described in section III-A, we ended up with $107,475$ sentences to be analyzed.

C. Word Embedding and Event-Code Classification

As introduced in Section III-C, we developed an algorithm using word embedding technique to recommend political interaction type through event codes for new found candidate patterns. In the next paragraphs we describe the empirical experiments we performed in order to desgin the algorithm and to find the best arguments for them.

We start by analyzing the word embedding models available as external resources. The initial idea is to identify a model that better suits to our data and how is the more appropriate way for using it.

Therefore our first experiment consists on randomly splitting the CAMEO verb dictionary in two disjoint groups with patterns and their respective event codes. The first group representing 66% of the dictionary was used as training data for classifying the event code of the second group, which simulates the candidate patterns. At this moment, since were are trying to identify the best model, we initially performed the experiment running the algorithm described in Algorithm 5, which is actually a specific case of Algorithm 4 when $k = 1$.

The results for our experiment is shown in Table III, which simply expresses the accuracy obtainined for each model. Overall we tested six models, five of them from GloVe (Global Vectors for Word Representation) [44] and one from Google [32].

In order to try to improve our results of how using the word embedding models, we focused on analyzing the wrong labeled cases. So, as our next experiment we kept the k most similar values for each one of the testing patterns. Then we compute the average of the variance of the top k similarity values for the portion we assigned correct label versus the portion we assigned wrong label. Table IV presents the results of this analysis when using glove-wiki-gigaword-100 model.

Table IV shows that the mean of the variance of the k lasgest values of similarity measured for the wrong labeled patterns is much larger than the same metric for the correctly labeled patterns. It intuitively may indicate that the wrongly labeled vectors usually are between more than one clusters of root-codes in the vector space. Besides, note that the more we increase the value of k, the lower the difference between the mean of variances of k largest similarity values for correctly and wrongly labeled verb-patterns.

```
Algorithm 5: Testing Models Accuracy
input  : Lists training and testing groups of
         existing CAMEO verb-patterns, word
         embedding model model.
output : Accuracy of the model.
$count \leftarrow 0$
$correct \leftarrow 0$
foreach $test$ in $testing$ do
    $max\_sim \leftarrow 0$
    $closest\_event\_code \leftarrow 0$
    $test\_vec \leftarrow model.vec(test.pattern)$
    foreach $train$ in $training$ do
        $train\_vec \leftarrow model.vec(train.pattern)$
        $curr\_sim \leftarrow COS\ (test\_vec,\ train\_vec)$
        if $curr\_sim > max\_sim$ then
            $max\_sim \leftarrow curr\_sim$
            $closest\_event\_code \leftarrow p.code$

    $count \leftarrow count + 1$
    if $closest\_event\_code = testing.code$ then
        $correct \leftarrow correct + 1$
$acc \leftarrow (correct/count)$
return $acc$
```

TABLE III
WORD EMBEDDING MODELS COMPARISON.

Models	Accuracy
glove-wiki-gigaword-50	0.6272
glove-wiki-gigaword-100	**0.6583**
glove-wiki-gigaword-200	0.6272
glove-wiki-gigaword-300	0.5695
GoogleNews-vectors-negative300	0.6052

TABLE IV
VARIANCE OF THE k LARGEST SIMILARITY VALUES MEASURED FOR CORRECTLY AND WRONGLY ROOT CODE CLASSIFICATIONS.

Group	Mean of variance of k largest similarities						
	k=3	k=5	k=7	k=9	k=11	k=13	k=15
Correctly Labeled	5.16	8.35	9.24	9.83	10.42	11.05	12.04
Wrongly Labeled	14.75	16.92	18.47	20.27	21.20	22.04	22.74

Therefore, it seems reasonable applying nearest neighborhood idea in our algorithm, so we can assign the root code on candidate patterns based on the root code that most frequently appears among those k most similar existing patterns. To confirm our hypothesis, we run the empirical experiment shown in Table V, which measures the accuracy for all the models previously evaluated, applying nearest neighborhood method for many values of k.

Considering the results observed in Table V, we decided to use glove-wiki-gigaword-100 word embedding model with $k = 13$ for the root code recommending procedure described in Algorithm 4.

TABLE V
ACCURACY FOR EACH MODEL CONSIDERING K NEAREST NEIGHBORS.

Models	k=1	k=5	k=9	**k=13**	k=17	k=21
gigaword-50	62.72	68.66	69.47	69.99	69.7	68.03
gigaword-100	65.83	72.76	74.78	***75.12***	74.37	73.86
gigaword-200	62.72	69.24	74.09	74.37	74.49	73.97
gigaword-300	56.95	65.26	72.01	73.34	74.89	73.8
GoogleNews	60.52	67.85	72.34	72.93	73.39	74.12

D. Sample of Found Candidate Verb-Patterns

In next paragraphs we close the experiments section by presenting some examples of patterns output from our system and the corresponding root event code recommended for them.

We start by showing a small sample of found candidate patterns for which the recommended root code seem to be correct.

Candidate Pattern: *[accept, resignation]*
Recommended Root Code: *08 - YIELD*
Sample of sentences where it appears:
Example 1: *Afghan president Asharf Ghani has accepted the resignation of the country's national security adviser, Mohammed Haneef Atmar, and replaced him with Afghan Ambassador to the U.S. Hamdullah Mohib.*
Example 2: *Shah Hussain Mortazavi, a spokesman for the Afghan leader, confirmed that Mr. Ghani had accepted the resignation of the national security adviser, Hanif Atmar, and appointed Hamdullah Mohib, the current ambassador to the United States, as his replacement.*

Candidate Pattern: *[make, disclosure]*
Recommended Root Code: *01 - MAKE PUBLIC STATEMENT*
Sample of sentences where it appears:
Example 1: *Buhari made this disclosure at the FOCAC round table meeting, attended by African leaders and Chinese President Xi Jinping.*
Example 2: *The International Committee of the Red Cross (ICRC) made a grisly disclosure about the humanitarian disaster going on in Nigeria as a result of the extremist campaign of terrorist and violent groups like Boko Haram, as well as the farmers-cattle herders' clashes.*

Following, we show an example of found candidate patterns for which the recommendation is questionable, or do not seem to be quite correct for all scenarios.

Candidate Pattern: *[drive, wedge]*
Recommended Root Code: *19 - FIGHT*
Sample of sentences where it appears:
Example 1: *Undoubtedly, Pyongyang will drive a wedge between the US and South Korea if their relations show cracks.*
Example 2: *Two Russian intelligence officers had been charged with attempted murder, the first criminal charges in*

a case that has driven a deep wedge between Russia and the West.

Finally, below we have an example where the recommendation of the root code was surely wrong.

Candidate Pattern: *['seize', 'opportunity']*
Recommended Root Code: *09 - INVESTIGATE*
Sample of sentences where it appears:

Example 1: *Majlis Speaker Ali Larijani said on Wednesday that Tehran seizes any opportunity to broaden relations and cooperation with Russia and Belarus.*

Example 2: *Mr Erdogan sees a rare moment when the Saudis are on the back foot and is seizing an opportunity to diminish a man labelled by a pro-government columnist here an enemy of Turkey.*

Another interesting observation we had during our computational experience was the effect of changing the thresholds while applying the apriori algorithm. By making the support and confidence thresholds more flexible, we obtain more general patterns like some phrasal verbs not yet coded in CAMEO verb-pattern dictionary. Examples of these cases are provided below.

Candidate Pattern: *[fend, off]*
Sample of sentences where it appears:

Example 1: *US launches airstrikes to help fend off major Taliban assault on Afghan city.*

Example 2: *Saudi Arabia is likely to successfully fend off challenges to its leadership of the Muslim world.*

Candidate Pattern: *[pave, way]*
Sample of sentences where it appears:

Example 1: *Russian Foreign Minister Sergei Lavrov is in Turkey to pave the way for a summit on Syria involving both countries as well as France and Germany.*

Example 2: *An Iranian delegation has traveled to Syria to pave the way for a transformation in their economic ties during the post-war era in the Arab country.*

Candidate Pattern: *[ramp, up]*
Sample of sentences where it appears:

Example 1: *U.S. President Donald Trump ramped up his attack on Turkey by raising steel and aluminium tariffs to 50 percent and 20 percent respectively.*

Example 2: *U.S. President Donald Trump is prepared to quickly ramp up a trade war with China and has told aides he is ready to impose tariffs on 200 billion more in Chinese imports.*

V. Future Works and Conclusions

In this paper we introduced the Automated Verbal-Pattern Extraction system as a tool for pottentially extending the CAMEO verb dictionary. We discussed about each step performed by the proposed system, presenting details of how candidate patterns and their corresponding classification of political interaction are extracted from news articles given as input.

The computational experiments performed in Section IV presented optimistic results in terms of candidate patterns found and its potential usage for extending the CAMEO verb-pattern dictionary.

In future works, we intend to either design or apply existing political science domain specific word embeddings models in order to improve the accuracy of the Event Code Recommendation step of the system. Besides, performing more extensive computational experiments using larger amount of data is absolutely part of our future works agenda. Finally, as we work with Unversal Dependencies tools, we intend to expand all the exercises performed in English for other languages like Spanish and Arabic.

Acknowledgements

This material is based upon work supported by NSF award numbers: DMS-1737978, DGE 17236021. OAC-1828467; ARO award number: W911-NF-18-1-0249; IBM faculty award (Research); and NSA awards.

References

[1] E. Boschee, J. Lautenschlager, S. O'Brien, S. Shellman, and J. Starz, "Icews automated daily event data," 2018. [Online]. Available: https://doi.org/10.7910/DVN/QI2T9A

[2] National Consortium for the Study of Terrorism and Responses to Terrorism (START), "Global terrorism database," 2018. [Online]. Available: https://www.start.umd.edu/gtd

[3] S. Salam, P. Brandt, J. Holmes, and L. Khan, "Distributed framework for political event coding in real-time," in *Proceedings of the 2018 conference on Electrical Engineering and Computer Science (EECS)*, 2018.

[4] T. Al-Khateeb, M. M. Masud, L. Khan, C. Aggarwal, J. Han, and B. Thuraisingham, "Stream classification with recurring and novel class detection using class-based ensemble," in *2012 IEEE 12th International Conference on Data Mining.*, 2012, pp. 31–40.

[5] M. M. Masud, J. Gao, L. K. J. Han, and B. Thuraisingham, "Classification and novel class detection in data streams with active mining," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 2010, pp. 311–324.

[6] M. A. Awad and L. R. Khan, "Web navigation prediction using multiple evidence combination and domain knowledge," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, no. 6, pp. 1054–1062, 2007.

[7] S. Abrol and L. Khan, "Twinner: understanding news queries with geo-content using twitter," in *Proceedings of the 6th Workshop on Geographic information Retrieval*, 2010, p. 10.

[8] C. Breen, L. Khan, and A. Ponnusamy, "Image classification using neural networks and ontologies," in *Proceedings. 13th International Workshop on Database and Expert Systems Applications*, 2002, pp. 98–102.

[9] L. Wang, L. Liu, and L. Khan, "Automatic image annotation and retrieval using subspace clustering algorithm," in *Proceedings of the 2nd ACM international workshop on Multimedia databases*, 2004, pp. 100–108.

[10] M. Awad, L. Khan, F. Bastani, and I. Yen, "An effective support vector machines (svms) performance using hierarchical clustering," in *16th IEEE International Conference on Tools with Artificial Intelligence*, 2004, pp. 663–667.

[11] I. Yen, J. Goluguri, F. Bastani, L. Khan, and J. Linn, "A component-based approach for embedded software development," in *Proceedings Fifth IEEE International Symposium on Object-Oriented Real-Time Distributed Computing.*, 2002, pp. 402–410.

[12] S. Chandra, L. Khan, and F. B. Muhaya, "Estimating twitter user location using social interactions–a content based approach," in *2011 IEEE Third International Conference on Privacy, Security, Risk and Trust and 2011 IEEE Third International Conference on Social Computing*, 2011, pp. 838–843.

[13] M. M. Masud, Q. Chen, L. Khan, C. Aggarwal, J. Gao, J. Han, and B. Thuraisingham, “Addressing concept-evolution in concept-drifting data streams,” in *2010 IEEE International Conference on Data Mining*, 2010, pp. 929–934.

[14] M. M. Masud, T. M. Al-Khateeb, L. Khan, C. Aggarwal, J. Gao, J. Han, and B. Thuraisingham, “Detecting recurring and novel classes in concept-drifting data streams,” in *2011 IEEE 11th International Conference on Data Mining*, 2011, pp. 1176–1181.

[15] M. M. Masud, Q. Chen, L. Khan, C. Aggarwal, J. Gao, J. Han, A. Srivastava, and N. Oza, “Classification and adaptive novel class detection of feature-evolving data streams,” *IEEE Transactions on Knowledge and Data Engineering*, pp. 1484–1497, 2013.

[16] M. M. Masud, J. Gao, L. Khan, J. Han, and B. Thuraisingham, “Integrating novel class detection with classification for concept-drifting data streams,” in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 2009, pp. 79–94.

[17] M. M. Mohammad, C. Woolam, J. Gao, L. Khan, J. Han, K. Hamlen, and N. Oza, “Facing the reality of data stream classification: coping with scarcity of labeled data,” *Knowledge and information systems*, no. 1, pp. 213–244, 2012.

[18] M. M. Masud, J. Gao, L. Khan, J. Han, and B. Thuraisingham, “A multi-partition multi-chunk ensemble technique to classify concept-drifting data streams,” in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 2009, pp. 363–375.

[19] L. Wang and L. Khan, “Automatic image annotation and retrieval using weighted feature selection,” *Multimedia Tools and Applications*, no. 1, pp. 55–71, 2006.

[20] G. Lavee, L. Khan, and B. Thuraisingham, “A framework for a video analysis tool for suspicious event detection,” *Multimedia Tools and Applications*, no. 1, pp. 109–123, 2007.

[21] P. Parveen, J. Evans, B. Thuraisingham, K. W. Hamlen, and L. Khan, “Insider threat detection using stream mining and graph mining,” in *2011 IEEE Third International Conference on Privacy, Security, Risk and Trust and 2011 IEEE Third International Conference on Social Computing*, 2011, pp. 1102–1110.

[22] A. Haque, L. Khan, and M. Baron, “Sand: Semi-supervised adaptive novel class detection and classification over data stream,” in *THIRTIETH AAAI Conference on Artificial Intelligence*, 2016.

[23] P. Parveen, Z. R. Weger, B. Thuraisingham, K. Hamlen, and L. Khan, “Supervised learning for insider threat detection using stream mining,” in *2011 IEEE 23rd International Conference on Tools with Artificial Intelligence*, 2011, pp. 1032–1039.

[24] L. L. Khan and L. Wang, “Automatic ontology derivation using clustering for image classification,” in *Multimedia Information Systems*, 2002, p. 65.

[25] L. Khan and D. McLeod, “Audio structuring and personalized retrieval using ontologies,” in *Proceedings IEEE Advances in Digital Libraries 2000*, 2000, pp. 116–126.

[26] L. Khan, *Ontology-based information selection*. ProQuest Information and Learning, 2000.

[27] F. Luo and L. Khan, “Ontology construction for information selection,” *Technical Report*, 2002.

[28] L. Khan, D. McLeod, and E. Hovy, “Retrieval effectiveness of an ontology-based model for information selection,” *The VLDB JournalThe International Journal on Very Large Data Bases*, 2004.

[29] M. Solaimani, S. Salam, L. Khan, P. T. Brandt, and V. D’Orazio, “Repair: Recommend political actors in real-time from news websites,” in *2017 IEEE International Conference on Big Data (Big Data)*, 2017, pp. 1333–1340.

[30] P. A. Schrodt and D. V. Brackle, “Automated coding of political event data,” in *Handbook of computational approaches to counterterrorism*. Springer, 2013, pp. 23–49.

[31] J. Nivre, M. D. Marneffe, F. Ginter, Y. Goldberg, J. Hajic, C. D. Manning, R. T. McDonald, S. Petrov, S. Pyysalo, N. Silveira *et al.*, “Universal dependencies v1: A multilingual treebank collection.” in *LREC*, 2016.

[32] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, “Distributed representations of words and phrases and their compositionality,” in *Advances in neural information processing systems*, 2013, pp. 3111–3119.

[33] K. Golnabi, R. Min, L. Khan, and E. Al-Shaer, “Analysis of firewall policy rules using data mining techniques,” in *2006 IEEE/IFIP Network Operations and Management Symposium NOMS 2006*, 2006, pp. 305–315.

[34] V. A. Petrushin and L. Khan, *Multimedia data mining and knowledge discovery*. Springer, 2007.

[35] B. Thuraisingham, L. Khan, M. M. Masud, and K. W. Hamlen, “Data mining for security applications,” in *2008 IEEE/IFIP International Conference on Embedded and Ubiquitous Computing*, 2008, pp. 585–589.

[36] M. Masud, B. Thuraisingham, and L. Khan, *Data mining tools for malware detection*. Auerbach Publications, 2016.

[37] R. Agrawal, T. Imieliński, and A. Swami, “Mining association rules between sets of items in large databases,” in *Acm sigmod record*, vol. 22, 1993, pp. 207–216.

[38] J. Lu. Universal petrarch: Language-agnostic political event coding using universal dependencies. [Online]. Available: https://github.com/openeventdata/UniversalPetrarch

[39] D. J. Gerner, P. A. Schrodt, and O. Yilmaz, “Conflict and mediation event observations (cameo) codebook,” *Manuscript*, 2009. [Online]. Available: http://web. ku. edu/keds/data. dir/cameo. html

[40] L. Wang, L. Khan, and B. Thuraisingham, “An effective evidence theory based k-nearest neighbor (knn) classification,” in *2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, 2008, pp. 797–801.

[41] M. Straka, J. Hajic, and J. Straková, “Udpipe: Trainable pipeline for processing conll-u files performing tokenization, morphological analysis, pos tagging and parsing.” in *LREC*, 2016.

[42] L. Ou-Yang. Newspaper3k: Article scraping curation. [Online]. Available: https://newspaper.readthedocs.io/en/latest/

[43] J. Beieler. Web scraper. [Online]. Available: https://github.com/openeventdata/scraper

[44] J. Pennington, R. Socher, and C. Manning, “Glove: Global vectors for word representation,” in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543.