APART: Automatic Political Actor Recommendation In Real-time

Mohiuddin Solaimani[1](orcid.org/0000-0001-7049-581X), Sayeed Salam[1](orcid.org/0000-0002-7254-749X), Latifur Khan[1](orcid.org/0000-0002-9300-1576), Patrick T. Brandt[2](orcid.org/0000-0002-7261-7056), and Vito D'Orazio[2](orcid.org/0000-0003-4249-0768)

[1]Dept. of CS, [2]School of Economic, Political, and Policy Sciences
The University of Texas at Dallas
Email: (mxs121731, sxs149331, lkhan, pbrandt,dorazio)@utdallas.edu

Abstract. Extracting actor data from news reports is important when generating event data. Hand-coded dictionaries are used to code actors and actions. Manually updating dictionaries for new actors and roles is costly and there is no automated method. We propose a dynamic frequency-based actor ranking algorithm with partial string matching for new actor-role detection, based on similar actors in the CAMEO dictionary. This is compared to a graph-based weighted label propagation baseline method. Results show our method outperforms the alternatives.

1 Introduction

Political event data [17,6] are coded from news reports and take the form of "who-did/said-what to whom." Automated event coders (e.g., PETRARCH [2] or BBN Accent [8]) use manually-entered dictionaries to identify actions and actors, and assign roles (e.g., government employee, media, etc.). The set of actions or verbs is rather finite and matched to CAMEO [16], but the set of nouns for actors and roles is large and constantly changing. Accurate and timely event data coding thus needs a real-time system to detect new actors and roles.

Designing a dictionary update system poses several challenges. First, actors may have multiple aliases: 'Barack H. Obama', or 'President Obama'. Second, the roles of actors change over time: Shimon Peres served multiple Israeli political roles. Finally, processing a large volume of international news articles demands scalable, distributed computing to detect this. We develop a real-time, distributed recommendation framework to identify actors and roles. We gather international news articles and pre-process them via Stanford CoreNLP and PETRARCH to extract actor data. Next, an unsupervised ranking algorithm recommends new actors and roles. A graph-based, weighted label propagation [11] actor-role recommendation method is implemented as a baseline.

We make three contributions: 1) is a novel time frequency- and window-based unsupervised new actor and role recommendation technique with alias actor grouping; 2) is a scalable real-time framework for coding actors; 3) is an improvement over a graphical propagation based actor-role recommendation.

2 Background

The first machine coder for event data was introduced by [15] anad was then developed into TABARI [14]. The DARPA-funded Integrated Crisis Early Warning System builds on this [12] and now provides global event data from 1995 [7]. Schrodt and Van Brackle [17] illustrate generating events from news texts. This earlier work focuses on event data generation and analysis, not incorporating dynamic dictionaries. Beieler, et al. [6] note that an event data challenge is manually developing CAMEO dictionaries with new entries. Relatedly, Saraf, et al. [13] show a recommendation model to detect reports of civil unrest.

Dynamically building actor dictionaries uses several tools: **Stanford CoreNLP** is a tool to annotate text with part-of-speech (POS) taggers and named entity recognition (NER), etc. [3]. **CAMEO** (Conflict and Mediation Event Observations) codes events, including actor specifications, to record political events [16]. **PETRARCH** (A Python Engine for Text Resolution And Related Coding Hierarchy) is a program that takes text in Penn Tree format [4] from CoreNLP and generates CAMEO-coded events [2]. **Apache Spark** is an open-source, distributed framework for data analytics that avoids the I/O bottleneck of the conventional two-stage MapReduce programs [1].

3 Framework

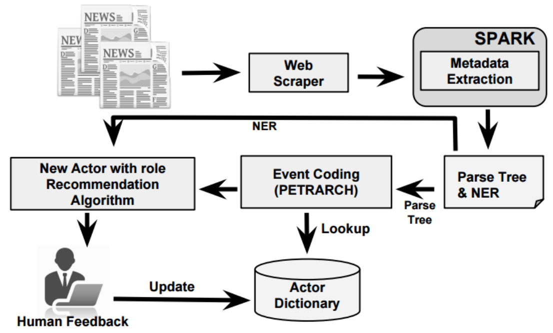


Fig. 1: Framework for real-time new political actor recommendation

Figure 1 shows the actor recommendation framework, an extension of [18]. A web-scraper [5] collects news periodically from about 400 RSS Feeds and extracts the main content which is then shipped through Apache Kafka to Apache Spark-based processing modules [18]. In the data processing unit, CoreNLP parses the reports and extracts meta-data, a parse tree, and NER. PETRARCH codes events from the parse tree, while New Actor uses NER and PETRARCH output. It captures PETRARCH generated events where political actors are unknown, and crosschecks it with NER to find new actors. Using our New Actor algorithm, new actors and roles are recommended. We provide a GUI and dashboard so that users can validate recommended actors and their roles for dictionary updates.

4 Recommending a new actor and role

Actors: The initial list of potential new actors and roles are those where PETRARCH suggests an event, but the actor is not present in CAMEO. From this list, we group the variations of an actor's name under a single actor identity (e.g., *Barack H. Obama* for *President Obama*, *Barack Obama*, etc.). Several similarity measures are used, such as Levenshtein Distance [10] and MinHash [9] to group the name variations. Each of these methods requires a similarity threshold sim_{th}. A score is generated for each actor via the following equation: $rank(a) = \sum_{d \in D} tf(a,d) \times df(a,D)$. The term frequency, $tf(a,d)$, shows the frequency of an Actor a in Document d. We use document frequency $df(a,D)$ to show the time window of document set D, where Actor a appears at least once. A buffered time window W of length L is maintained to find N actors, which are merged with the previous window's list. The rank statistics are updated after the merge. After L windows, new actors and their roles are recommended if their occurrences in the L_{th} buffered window exceed the threshold TH.
Roles: Role recommendations are based the similarity of new actors to those with whom they interact. Actors from one country or government are more likely to interact with ones from the same country or government. When a new potential actor is identified, the top M most frequent roles are recommended from among the co-occurring roles of any co-occurring actors. Co-occurring actors are those actors that appear in the same document. When an actor appears in two documents, the roles of all co-occurring actors in both articles are included. When a co-occurring actor has multiple roles, we include each.
Recommendations: In each time window all articles are scanned for the potential actor list with rankings and role recommendations. If an actor comes in the top N rankings in multiple time windows, s/he has high probability of being a new political actor. For the threshold TH, new actors are those that appear in 5 or more windows, but those who appear less than 3 are discarded.

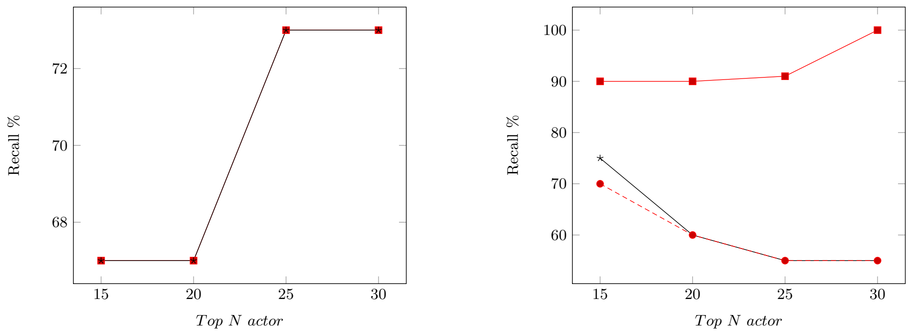
5 Experiments

To evaluate the framework, in $L = 9$ time windows at 2 hour intervals, newly published news articles from about 400 RSS feeds were scraped. For this empirical study, we estimate thresholds for edit distance and min-hash based methods to be $sim_{th} = .75$ and $sim_{th} = .45$, respectively, based on minimizing false positives using known alias groupings provided by the CAMEO actor dictionary.

A graph-based role detection technique is the baseline comparison. This models the interactions between actors using a Graph, $G = (V, E)$, where V is the set of existing and recommended actors and E contains their edge interactions to infer roles using the weighted label propagation technique[11]. The co-occurrence of two actors in the same document is an interaction and the frequency of co-occurring actors is their weight. After formulating the graph, we begin the weighted label propagation algorithm. The existing actors, those that are in CAMEO, have their roles as the labels. The recommended actors, those not in CAMEO, begin with an *empty* label.

We next eliminate some well known actors from the CAMEO actor dictionary and try to recover them. This experiment removed 5, 10, and 15 actors, but due to space we only present the results from removing 15. The results are consistent when removing 5 and 10. Since PETRARCH will not code events for the deleted actors, they are recommended as if they are newly discovered. The recall is the percentage of the removed actors then recommended by our algorithm. Precision is not computed because all the recommended actors are political.

Performance Evaluation. Fig. 2(a) plots recall when grouping actor aliases using edit distance and MinHash. As the top N recommended actors increase, the recall for retrieving a deleted actor increases. Of the 15 actors removed from CAMEO, 12 are included in the top 25 recommendations. Although not apparent in Fig. 2(a), for closely matched actor aliases like ‘Donald Trump’ and ‘Donald J. Trump’, edit distance performs slightly better than MinHash. In Fig. 2(b), examines recall with 5 suggested roles. The suggestion is a success if one of the recommended roles is the true role in CAMEO. For Exact match, the recommended role must be exactly identical to the true role. For edit distance and MinHash, partial matching is allowed using substrings (e.g., suggesting USA for USAGOV is a success) and thresholds, to allow for roles that are near the true role but not identical. While edit distance is the better method, MinHash and Exact show good recall.



(a) Deleted actor = 15

(b) Deleted actor = 15

Fig. 2: Performance for (a) Actor recommendation and (b) Role recommendation. Recall: Edit distance —■—, MinHash —⋆—, Exact match - -●- - (b only)

Word2Vec. We have experimented with using Word2Vec for role recommendations by training a model to predict co-occurring actors. Specifically, for each document in each of the L windows, we extract all actors recognized by NER and input that actor list as an observation for training the Word2Vec model. When the New Actor algorithm proposes a new actor, the Word2Vec model predicts co-occurring actors, extracts their roles, and proceeds with role recommendation. In this way, it is possible that an actor that does not co-occur, in the strict sense of appearing in the same article, may be among the list of predicted co-occurring actors whose roles are extracted. After applying this method, the suggested roles had less than 5% recall due to poor performance by the co-occurring actor predictions. We suspect this was due to the lack of alias groupings as a pre-processing step, and leave this for future research.

Graph-based Comparison. The frequency-based approach for role recommendation outperforms the graph-based approach by a considerable margin. We fix the values of the number of recommended actors per window at $N = 20$, and again delete 15 actors from CAMEO. The frequency-based approach outperforms the graph-based one for each similarity measure: 90 to 27 for edit distance, 60 to 20 for MinHash, and 61 to 20 for Exact. The frequency-based approach considers roles from existing actors in the CAMEO dictionary, while the graph-based approach considers roles from neighbors who are either existing or new actors. Thus the error with one role assignment can propagate to others.

Validation. As validation of the system, Table 1 shows the top recommended actors across all windows for the two string similarity measures, using the same thresholds and time-windows as in the experiments. Both methods detect similar roles for identical actors, but suggest different actor lists. Given that we are building a recommendation system for expanding a political actor dictionary, we can see that the new actors are quite appropriate—Donald Trump, Amir Sheik Sabah, Rodrigo Duterte, and others are prominent government officials.

Table 1: List of recommended actors with their roles

Edit Distance		**MinHash**	
Actor	**Top 3 roles**	**Actor**	**Top 3 roles**
DONALD TRUMP	USA, USAELI, LEG	SEDIQ SEDIQQI	AFG, PPL, UAF
SEDIQ SEDIQQI	UAF, AFG, PPL	DONALD TRUMP	USA, USAELI, LEG
AMIR SHEIK SABAH AL-AHMAD AL-JABER AL-SABAH	SAUMED, SAUGOV, KWTMEDGOV	AMIR SHEIK SABAH AL-AHMAD AL-JABER AL-SABAH	SAUMED, SAUMEDGOV, KWTMEDGOV
LYNNE O' DONNEL	AFG, AFGUAF, PPL	RODRIGO DUTERTE	PHLGOV, GOV, PHL
RODRIGO DUTERTE	PHLGOV, GOV, PHL	ANTIO CARPIO	JUD, PHL, CRM

6 Conclusion and future work

Political actor dictionaries are integral to political event data coding. We address the problem of detecting and recommending new actors and their roles in real-time. A Spark-based framework with unsupervised rankings of new actor aliases on a periodic basis is proposed. Currently, this is only to find new actors, but it can be extended to recommend new events in the CAMEO verb dictionary.

Acknowledgments: Support from the National Science Foundation (NSF) SBE-SMA-1539302, CNS-1229652, and SBE-SES-1528624; and the Air Force Office of Scientific Research (AFOSR): FA-9550-12-1-0077. Any opinions, findings, and conclusions or recommendations expressed here are those of the authors and do not necessarily reflect the views of the NSF or the AFOSR.

References

1. Apache Spark. [Online]. Available: http://spark.apache.org/.
2. Petrarch. `http://petrarch.readthedocs.org/en/latest/`.
3. Stanford CoreNLP. `"http://nlp.stanford.edu/software/corenlp.shtml"`.
4. The Penn Treebank Project. `"https://www.cis.upenn.edu/~treebank/"`.
5. Web Scraper. `http://oeda-scraper.readthedocs.io/en/latest`.
6. J. Beieler, P. T. Brandt, A. Halterman, P. A. Schrodt, and E. M. Simpson. Generating political event data in near real time: Opportunities and challenges. *Computational Social Science: Discovery and Prediction. ed. by R. Michael Alvarez, Cambridge, Cambridge University Press*, pages 98–120, 2016.
7. E. Boschee, J. Lautenschlager, S. O'Brien, S. Shellman, J. Starz, and M. Ward. ICEWS Coded Event Data, 2016.
8. E. Boschee, P. Natarajan, and R. Weischedel. Automatic extraction of events from open source text for predictive forecasting. In *Handbook of Computational Approaches to Counterterrorism*, pages 51–67. Springer, 2013.
9. A. Z. Broder. On the resemblance and containment of documents. In *Compression and Complexity of Sequences 1997. Proceedings*, pages 21–29. IEEE, 1997.
10. V. I. Levenshtein. Binary codes capable of correcting deletions, insertions and reversals. In *Soviet physics doklady*, volume 10, page 707, 1966.
11. H. Lou, S. Li, and Y. Zhao. Detecting community structure using label propagation with weighted coherent neighborhood propinquity. *Physica A: Statistical Mechanics and its Applications*, 392(14):3095–3105, 2013.
12. S. O'Brien. Crisis early warning and decision support: Contemporary approaches and thoughts on future research. *International Studies Review*, 12(1):87–104, 2010.
13. P. Saraf and N. Ramakrishnan. EMBERS autogsr: Automated coding of civil unrest events. In *ACM SIGKDD, San Francisco, CA, USA, August 13-17, 2016*, pages 599–608, 2016.
14. P. A. Schrodt. *TABARI: Textual Analysis By Augmented Replacement Instructions*, 2009. `http://eventdata.psu.edu/tabari.html`.
15. P. A. Schrodt, S. G. Davis, and J. L. Weddle. Political science: Keds-a program for the machine coding of event data. *Social Science Computer Review*, 12(4):561–587, 1994.
16. P. A. Schrodt, D. J. Gerner, and Ö. Yilmaz. Conflict and mediation event observations (CAMEO): An event data framework for a post Cold War world. In J. Bercovitch and S. Gartner, editors, *International Conflict Mediation: New Approaches and Findings*. Routledge, New York, 2009.
17. P. A. Schrodt and D. Van Brackle. Automated coding of political event data. In *Handbook of Computational Approaches to Counterterrorism*, pages 23–49. Springer, 2013.
18. M. Solaimani, R. Gopalan, L. Khan, P. T. Brandt, and B. Thuraisingham. Spark-based political event coding. In *BigDataService*, pages 14–23. IEEE, 2016.