Automated Dictionary Generation for Political Event Coding: Online Appendix

BENJAMIN J. RADFORD

Actor Classification

In the CYLICON application, ADG was shown to produce dictionaries capable of event-coding a corpus in a novel domain with minimal researcher input. Now I will demonstrate that an extension to the ADG process can be used to not only identify new terms and phrases, including actors, but to also associate actors with countries. It is assumed, for this task, that a list of actors and a list of countries are available to the researcher, possibly as the result of ADG, but that the associations between actors and countries are unknown. Actors are then classified by country using both unsupervised and supervised approaches. As with the ADG method presented above, word2vec is central to this task.

As with ADG, a word2vec model is first trained on a text corpus. In the fully unsupervised case, the actors' learned representations from the word2vec model are compared to the learned representations associated with locations (e.g. city, province, and country vectors). The k-nearest neighbors algorithm is then used to classify actors based on proximity to location vectors. In the supervised approach, it is assumed that the researcher has access to an existing country-actor dictionary that is in need of updating. A random forest is used to build a predictive model that maps actor vectors to predicted classes (i.e. country labels). Both methods are evaluated against a "ground truth" dataset.

TABLE 1 *ICEWS data summary.*

	Documents	Learned words and phrases
ICEWS90	570,488	682,324
ICEWS183	1,210,483	1,136,519
ICEWS365	2,441,345	1,793,776
ICEWS730	5,058,635	2,605,372

To build training and test sets, the Phoenix country-actor dictionary is compared to the word2vec model vocabulary and those actors that are present in both the word2vec model and the dictionary are selected.[1] Because these actors are already classified by country in the Phoenix country-actor dictionary, a "ground truth" actor-country dataset can be produced for evaluation purposes. When actors are assigned multiple codes over time in the Phoenix dictionary, the first code, generally the earliest in time, is selected as that actor's country affiliation. It is uncommon for a single actor to be assigned to more than one country.

Word embedding models are sensitive to both corpus size and corpus pre-processing techniques. For this reason, multiple corpora are selected to train word2vec models and the results are compared. A model provided by Google and trained on the entirety of Google News content is used to prove the viability of this method. Additionally, a series of models trained on news data from the ICEWS dataset are examined to test the feasibility of this method for researchers unable to obtain a corpus as thorough as Google's.[2]

Google's model, here referred to as *GoogleNews*, was trained on 100 billion words and resulted in a dataset describing 3 million unique words and phrases as vectors of length 300 (Google 2015). The ICEWS text corpus for 2013 and 2014 is used to train

[1] The Phoenix dictionaries are chosen over the ICEWS dictionaries for consistency with the rest of the paper.

[2] For the ICEWS-based models, the `min_count` parameter is set to 3 and the vector `size` parameter is set to 300. Other parameters are left at the default values.

four models.[3] These models cover 90 days of news, 183 days of news, 365 days of news, and 730 days of news and are referred to as *ICEWS90*, *ICEWS183*, *ICEWS365*, and *ICEWS730*, respectively. Table 1 summarizes the ICEWS data. The ICEWS corpus consists of only those news stories that contain events from the ICEWS event dataset. ICEWS events are CAMEO-coded and therefore represent instances of political conflict and cooperation. ICEWS texts are drawn from roughly 300 different publishers (Raytheon BBN Technologies 2015).

For illustration purposes, the word vectors for actors in the ICEWS730 corpus are plotted in Figure 1. t-distributed Stochastic Neighbor Embedding (t-SNE) is used to project the word vectors into a two dimensional space while preserving local distances between neighboring points. t-SNE, a non-linear dimensionality reduction technique, was introduced by Maaten and Hinton (2009) and implemented for R by Radford (2017). All actors from the top ten countries (by number of unique actors) are colored by their associated country. The figure shows clustering by country and some separation between the actors of different countries. Note that t-SNE is fully unsupervised and that any clustering by country observed in the plot is a function of the word vectors alone.

TABLE 2 *Model performance on classifier task.*

Model	Actors	Locations	Countries	Accuracy				
				k=1	k=3	k=5	k=7	RF
ICEWS90	2884	2271	182	0.25	0.27	0.29	0.30	0.26
ICEWS183	3890	2447	182	0.29	0.31	0.34	0.35	0.32
ICEWS365	4764	2630	182	0.32	0.34	0.37	0.38	0.35
ICEWS730	5096	2756	183	0.37	0.39	0.42	0.43	0.43
GoogleNews	4932	2616	183	0.58	0.62	0.64	0.64	0.54

[3] Unfortunately, the ICEWS corpus is only available to researchers involved with the ICEWS project. The author only had access to texts that produced events in the ICEWS event dataset, not the full corpus of texts processed by ICEWS. Therefore, no “negative” ICEWS texts are included here.

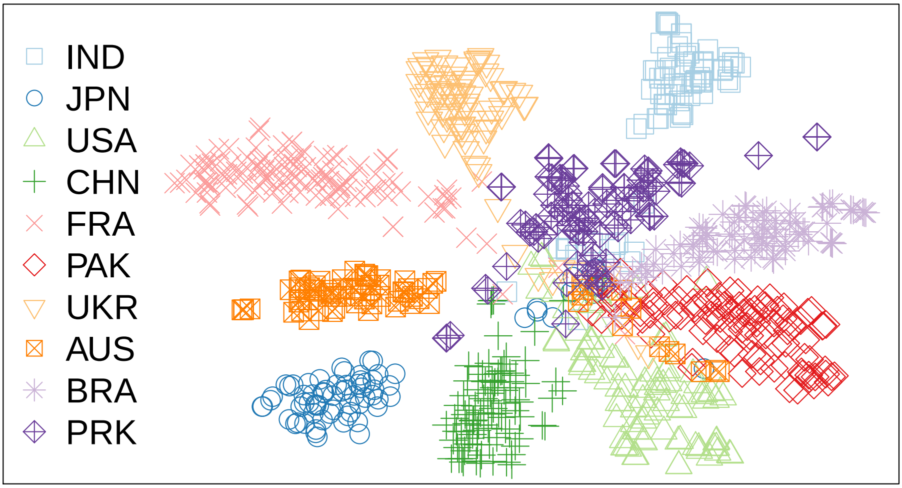


Figure 1. *Embedding of actor vectors colored by ground truth country.*
Note: t-SNE projection, *ICEWS730* corpus.

Unsupervised Actor Classification

In the first classification task, a set of location names and a set of actors to classify by country are given, but no training data are available. The objective is to simulate the process of ingesting news and producing country-actor dictionaries for event coding without human intervention. Actor names are assumed to have been extracted using ADG or NER while location names are assumed to be available from, for example, a gazetteer.

A cosine similarity matrix is constructed between the vectors associated with the actors' names and the vectors associated with location names. Each actor is then assigned to the country associated with the location whose vector is most similar, by cosine similarity, to the actor's vector. The process is tested for all five word2vec models and the results (test-set classification accuracy) are shown in Table 2. Additionally, k-nearest neighbors voting is applied for odd values of k between 3 and 7.

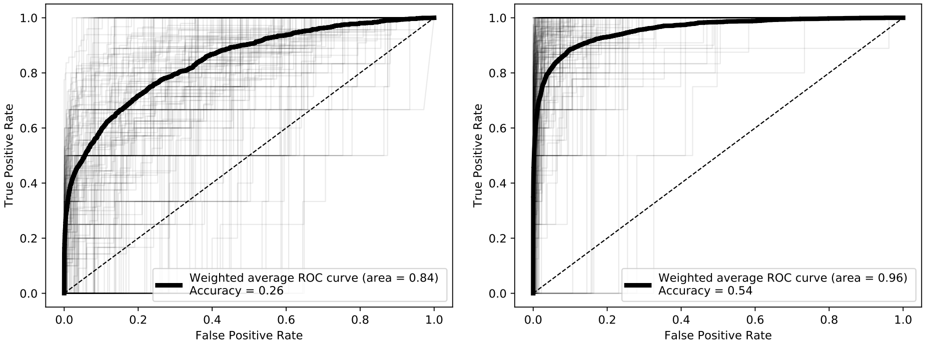
All five models substantially outperform weighted random assignment in classifying actors.[4] The worst performing model correctly classifies 25% of the actors. Models trained on larger corpora achieve better performance than those trained on smaller corpora and k-nearest neighbors voting improves accuracy as k increases. The model based on *GoogleNews* correctly classifies between 58% and 64% of all actors in the Phoenix country-actor dictionary. The next highest scores are achieved by *ICEWS730*, correctly classifying between 37% and 43% of actors. Interestingly, more Phoenix actors and locations are identified in the *ICEWS730* model's vocabulary than in *GoogleNews*'s. This is likely due in large part to the ability of the researcher to match data pre-processing procedures between the dictionary and the *ICEWS* models.[5] Previous work in this area has focused on event-level location prediction and is therefore not directly comparable to the actor-level location prediction results reported here. Lee, Liu, and Ward (2016) report up to 84% accuracy in the supervised task of predicting which location term in a news story describes the location in which an event occurred (as opposed to locations that are referenced for background or context). An unsupervised method for event-level location prediction, Mordecai, is introduced by Halterman (2017), but performance metrics are not yet available.

Supervised Actor Classification

The supervised test assumes that the researcher has access to a training sample of classified terms or phrases. In this case, the set of actor-location pairs from the Phoenix country-actor

[4] Simulations show that random assignment, weighted by class proportion, correctly classifies 1% of actors, on average.

[5] The discrepancy in performance between *GoogleNews* and *ICEWS730* could be the result of corpus size or model training parameters.



(a) ICEWS90 *corpus.*

(b) GoogleNews *corpus.*

Figure 2. Multiclass ROC plots for random forest classifier. Each thin line represents the ROC curve for a single country classifier evaluated in one-versus-all fashion. The bold line represents the weighted average ROC curve for all countries.

dictionary is split in half to produce a training set and a test set. The split is performed via simple stratified random sampling such that the distribution of labels (countries) is matched across the training and test set. This has the effect of preventing situations in which the training set excludes entire categories. Actors from countries that appear only once in both the word2vec model and the country-actor dictionary are excluded as they cannot be represented in both the training and test sets. A random forest model is then learned on the training set and used to predict labels for the held-out test set.[6] Actor word vectors constitute the input features of the random forest model that attempts to predict country affiliation.

Because it is possible to compute class membership probabilities from a the random forest model, receiver operating characteristic (ROC) plots can be drawn to visualize

[6] *scikit-learn* is used to learn the random forest model (Pedregosa et al. 2011). Parameters are left at their default values with the exceptions of `max_depth` 20 and `n_estimators` 1000.

model performance. The worst- and best-case examples are depicted in Figures 2a and 2b. The accuracy of the random forest on each word2vec model is listed in the *RF* column of Table 2. Despite the use of ground truth labels, the accuracy of the random forest model largely mirrors the accuracy of the proposed unsupervised classification approach. The random forest successfully classifies 26% of actors on the smallest corpus and 54% of actors on the largest corpus, *GoogleNews*. The models' corresponding weighted AUC values, the areas under the ROC curves, are 0.84 and 0.96, respectively. Unlike accuracy, the AUC measures classification performance of a model across all possible thresholds; an AUC of 1 indicates perfect classification while an AUC of 0.5 is equivalent to random classification. High AUC values in the range seen here, above 0.8, are indicative of strong classification performance when averaged across all possible thresholds.[7]

References

Google. 2015. "word2vec." `https://code.google.com/p/word2vec/`.

Halterman, Andrew. 2017. "Mordecai: Full Text Geoparsing and Event Geocoding." *The Journal of Open Source Software* 2 (9). doi:`10.21105/joss.00091`.

Lee, Sophie J., Howard Liu, and Michael D. Ward. 2016. "Lost in Space: Geolocation in Event Data." *arXiv:1611.04837* (November 14).

Maaten, Laurens van der, and Geoffrey Hinton. 2009. "Visualizing High-Dimensional Data Using t-SNE." *Journal of Machine Learning Research* 9 (Nov): 2579–2605.

[7] The threshold here being the minimum predicted probability required to assign an actor to a particular country.

Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, et al. 2011. “Scikit-learn: Machine Learning in Python.” *Journal of Machine Learning Research* 12:2825–2830.

Radford, Benjamin J. 2017. “mmtsne: Multiple Maps t-SNE.” `https://cran.r-project.org/web/packages/mmtsne/index.html`, *CRAN: The Comprehensive R Archive Network.*

Raytheon BBN Technologies. 2015. “BBN ACCENT Event Coding Evaluation.”

Post-processing and seed dictionaries

The following steps are taken to prune the extracted terms and phrases.

- Verb phrases must contain at least one verb (a word tagged VB_).
- Verb phrases must contain at least one verb above a certain inverse term-frequency value.[8]
- Verb phrases that include either of the words “no” or “not” are omitted.
- Verb phrases that end with “by” have “$” added to the end and “+” appended to the beginning in order to to switch the source-target actor. Single-word verb phrases are duplicated and have “by” appended to them in order to catch additional instances of target-verb-source. This rule is redundant due to internal detection of passive voice by PETRARCH but is kept for completeness.
- Agents and actors must contain at least one noun (a word tagged NN_).
- Agent and actor phrases that contain a verb are omitted.
- Duplicate phrases (those assigned to more than one category) are assigned strictly to that category with which they have the greatest cosine similarity.
- Words found in the synsets that appear in verb phrases are replaced in those verb phrases with the relevant synset category word.
- A minimum cosine similarity is chosen. 0.6 is used here. Terms and phrases that are less cosine-similar to the mean category vector than this value are discarded.

[8] A sample of 10,000 sentences of the corpus is transformed into a document-term matrix. Then, the term frequencies, f_{term}, are calculated as the percentage of documents that every word appears in. $1 - f_{term}$ produces the term weights. Verb phrases must contain a verb scored at 0.99 or above to be included. This has the effect of omitting phrases that include only high-frequency verbs such as “was,” “have,” and “been.”

TABLE 3 *Seed phrases for verb dictionary*

Verb Dictionary Seeds	
Category	Seed Phrase
DEFACED	DEFACED:O:VBD
PATCHED	PATCHED:O:VBD
INFILTRATED	BREACHED:O:VBD
LEAKED	LEAKED:O:VBD
PHISHED	PHISHED:O:VBD
DDOSED	DISTRIBUTED:O:VBN_DENIAL-OF-S . . .
INFECTED	INFECTED:O:VBD
VULNERABILITY	VULNERABILITY:O:NN
ARRESTED	ARRESTED:O:VBD
CENSORED	CENSORED:O:VBD
Agent and Actor Dictionary Seeds	
Category	Seed Phrase
HACKER	HACKER:O:NN
RESEARCHER	RESEARCHER:O:NN
WHISTLEBLOWER	WHISTLEBLOWER:O:NN
USERS	USERS:O:NNS
ANTIVIRUS	ANTIVIRUS:O:NN
Issue Dictionary Seeds	
Category	Seed Phrase
TOR	TOR:O:NNP
0DAY	0DAY:O:NN
HACKTIVISM	HACKTIVIST:O:NN
APT	APT:O:NN
DDOS	DDOS:O:NN
SOCIALENGINEERING	PHISH:O:VB
Synset Dictionary Seeds	
Category	Seed Phrase
HARDWARE	HARDWARE:O:NN
VIRUS	VIRUS:O:NN
COMPUTER	COMPUTER:O:NN
WEBASSET	WEBSITE:O:NN
SOFTWARE	SOFTWARE:O:NN