Taming the Firehose: Thematically summarizing very large news corpora using topic modeling [*]

Philip A. Schrodt
Parus Analytics LLC
Charlottesville, Virginia, USA
schrodt735@gmail.com

Version 1.0 : July 13, 2018

[*] Paper presented at PolMeth XXXV: 2018 Conference of the Society for Political Methodology, Brigham Young University, 18-21 July 2018. The opinions expressed herein are solely those of the author and do not reflect those of any of the various data projects with which the author has been associated now or in the past particularly those funded by the U.S. government. The programs used for both the data preparation and analysis are reasonably well documented and are available from the author. A link to the most recent version of this paper can be found at `http://eventdata.parusanalytics.com/papers.dir/automated.html`

Abstract

This paper instantiates the new "application-focused" format for PolMeth XXXV. The application in question is producing thematic chronologies from very large corpora of news texts (both native English and machine-translated Arabic) using a combination of political event data coding—specifically, a successor to the event coder used in the ICEWS project—and latent Dirichlet allocation (LDA) topic modeling as implemented through the open source program `gensim`. Because this is an applied project where the less-than-infinitely-patient end-users are looking for plausible and more or less distinct clusterings, rather than whatever dog's breakfast is produced by LDA, the two algorithmic challenges are reconciling the indeterminate outcomes from LDA (that is, due to numerical optimization over a high-dimensional surface characterized by many local optima, multiple runs produce different clusterings) and identifying similar clusters within a single runs. The system—which is entirely unsupervised after various pre-processing steps, including the use of an automated event coder similar to that used by the ICEWS project to restrict the corpus to sentences involving transactional events—is producing reasonably coherent (and consistent) results, with an interesting distinction between the results where the texts all involve a single country, where thematic clusters tend to align according to the foreign relations with other countries, and a clustering on a much larger corpus involving the entire Middle East, where the clusters are more behavioral. Issues remaining in the system are appropriate summarization—`gensim`'s function for this seems less than completely reliable—and differences induced by the very different stylistic characteristics of native English and translated Arabic.

1 *Ceci n'est pas un document universitaire*

In keeping with the Society for Political Methodology's long history of experimentation and innovation—a distinct contrast to the "conference like it's 1965!" approach of certain four-letter organizations which also have the word "Political" in their title[1]—the PolMeth XXXV call-for-papers[2] announced a set of format changes which departed from using only its traditional "long-form" single-paper/discussant format to including some of the more familiar 3 to 5 paper panels[3] and, most intriquingly, perhaps as a response to the near-collapse of the academic job market combined with the extraordinary private sector opportunities now available to ~~political methodologists~~ data scientists, also provided for the following option:

> Papers with a focus on application: these are papers that do not develop new methodology, and instead employ existing methods creatively to answer substantive questions

Well, y'all are going to open that barn door, this donkey is going to saunter through it.

So, my interpretation of "focus on application" is that this document is going to tell you how to do something useful. Well, at least something that someone has found sufficiently useful that they paid me to do it. What the presentation most definitely is *not* going to do is provide all of the boiler-plate expected of a work intended for eventual publication—following a lag of three to five years—in a paywalled journal.[4] Instead my exposition will focus on:

- Justifying the approaches I used and describing the applied niche I am trying to fill.
- Describing the advantages and disadvantages of the open source software package[s] I'm using in a manner that will allow you to decide whether it's worth making the "free as in puppy" investment in using them in your own projects.
- Describing the pre-processing pipeline in detail, since in data science one typically spends about 80% of a project getting the data to the point where it can be analyzed with sophisticated techniques, and pre-processing decisions can have substantial effects on the subsequent results.
- Describing the computational approach in sufficient detail that you can decide whether it is worth the effort to try to adapt my [sort of documented] open-source code or whether you'd be as well off just writing your own.

But as this is a work about a non-academic application of political methodology, it does not have the following ubiquitous features of academic works:

[1] One of the more notable being declaring independence from a certain four-letter organization which also has the word "Political" in its title.

[2] `https://www.cambridge.org/core/membership/spm/conferences/polmeth-2018`

[3] Five paper panels?!?... noooooooooo...

[4] For a more extended discourse on this process, see Footnote 8 in `https://asecondmouse.wordpress.com/2018/06/18/witnessing-a-paradigm-shift/`

- 35 pages with a 100-page web appendix, this despite the fact that Einstein managed to present the general theory of relativity in about 23 pages and no web appendix

- A massive bibliography of citations to paywalled journal articles dating back to the early Pleistocene. Pro-tip: *Paywalled journals are virtually inaccessible outside universities so by publishing in these venues you might as well be burying your intellectual efforts beneath a glowing pile of nuclear waste somewhere in Antarctica.* I will instead exclusively employ open, contemporary, web-based references. I will also assume that the reader has access to Google—which is to say, the reader is not working in a SCIF, at least at the moment—and can look up technical terms.

I do regret that in this instance—as is not infrequently the case for text-as-data work using large corpora of news sources—I can't provide replication data due to a variety of intellectual property restrictions, though I hope that I have described the data in sufficient detail that it would be straightforward to duplicate the method, if not the precise results.[5]

So unless you are comfortable with this "application" approach, this is not the paper you are looking for, move along, move along. Otherwise, we shall proceed.

2 The problem: "Drinking from a firehose"

The problem of "Drinking from a firehose"—having far too much information to effectively analyze—has been an issue for political analysts since the dawn of electronic communications, and has accelerated with the advent of the World Wide Web, particularly once almost every news source in world became available through web pages. Where analysts were once confined to analyzing just the information available through a few government-controlled sources such as the Soviet Union's *Pravda*[6] supplemented by [typically] wire service reports—even in the 1980s these were typically generating only about 2000 reports per day because they went through multiple layers of human editing—and capital city newspapers, possibly summarized in a few relevant mimeographed pages provided by the Foreign Broadcast Information Service and possibly further supplemented by selected tidbits from cable traffic from embassies, anyone with access to the internet can now access thousands of different local sources, typically down to the level of regional centers and often as local as small market towns, as well as a 24/7/365 flow of information from about a dozen major wire services.[7]

It is literally impossible for any analyst to process—or even read—all of this information, even when the topic of interest is a relatively small country such as Qatar or Yemen, the examples used here.[8] Consequently the issue of finding an automated method of summarizing

[5] Were sufficient time available, I could have done this with a somewhat more open data set such as the Linguistic's Data Consortium's (`https://www.ldc.upenn.edu/`) *Gigaword* news corpus, though even this requires quite some expense to access. Legally.

[6] The USSR's version of the Fox cable network.

[7] Ce n'est pas vraiment une phrase anglaise bien écrite.

[8] Cases where no credible local sources exist, such as North Korea, present an entirely different set of issues but are relatively rare. The Web-accessible local sources vary widely in their credibility and focus, and the influence by local economic elites on their content is usually at least as major a factor as centralized

these news streams, ideally in a fashion reasonably comparable to what would be produced by a human team with ample time and effort to study all of the stories, has long been a priority. Or a dream. Or hope. Or something like that. This has been idealized as the "analyst's workstation" and I'm familiar with proposals in more or less identical form which have been floating around since at least the 1980s "artificial intelligence" craze,[9] and just last week IARPA released a BAA for a more focused multi-lingual "small data" effort called BETTER on just this theme.[10]

Meanwhile, there are at least three common methods that don't work particularly well on this problem:

Boolean keyword searches: While readily available in the search engines of aggregators such as Lexis-Nexis and Factiva, these tend to produce a large number of false positives while also being insensitive to the use of synonymous terms, particularly across heterogeneous sources. Human-indexing of sources into categorical "subjects", which is done on some sources by some aggregators, helps but is only available on a relatively small number of fairly predictable topics—typically business-oriented—and depends on *a priori* typologies.

Example-based document retrieval: This is a very mature technology but has three weaknesses. First, generating sufficient examples for the method to work effectively may require a fairly substantial effort on the part of the analyst. Second and closely related, example-based systems presuppose the analyst already has a clear idea of what she is looking for, and are not a natural fit to exploratory work in a new corpus where the analyst is above all interested in new and unanticipated behaviors and issues. Third, document retrieval tends to work best on, well, documents, which are typically substantially longer than simple reports of interactions, which are frequently only a single sentence in length.

Event data: Political event data has, from its beginnings in the 1960s, been intended to provide succinct statistical summaries of interactions between political actors as a time series, and a variety of systems can now produce these data automatically in real time and at a very low cost. Unfortunately, as a number of projects which have tried to move these from the experimental to operational level have discovered, most political

government censorship but hey, that's why political analysts get paid the big bucks and are so consistently supported at the highest levels of government.

[9] But probably well before: Vannevar Bush's now legendary 1945 proposal on automated information retrieval (`https://en.wikipedia.org/wiki/Information_retrieval`) certainly was motivated by similar concerns. By the time I was involved in one such project in the early 1980s, the phrase "chronology generator" was used in the context of analytical tools, chronologies of who has been doing what to whom always having been the fundamental grist for a political analysis trying to ascertain what an actor might be doing next. The particular project I was involved with, funded by a major US ally, went absolutely nowhere, and from the perspective of 2018, we had nothing remotely close to either the data, hardware, or software tools to make a serious run at it. In another thirty years, of course, someone could almost certainly say exactly the same thing about the project I'm describing here. Probably more like in ten years.

[10] `https://www.fbo.gov/index?s=opportunity&mode=form&id=de8c791380ef142c367858542c74fab2&tab=core&_cview=1`. If you'd like to be a prime on this, please contact me: I've got a team with great ideas but not the 800-lb gorilla capabilities of primes.

analysts don't actually want *statistical* summaries—in fact that's usually the last thing they want except possibly as grist for a PowerPoint slide or two—and instead need the original texts. In addition, the most widely-used event data ontology, CAMEO, was not designed to capture general political events, and due to limitations in their dictionary updating, some systems are limited in their ability to pick up new political actors (for example ISIS and Boko Haram) until well after the actors have become important.

3 A partial solution: Topic Modeling

The limitations of the first two methods in a variety of information retrieval domains contributed by the late 1990s to the development of topic modeling, which attempts to abstract common themes from a set of documents based on word frequency and co-occurrence, but without the need for either examples or an *a priori* thematic typology. Interest in the approach increased dramatically with the development of the latent Dirichlet allocation (LDA[11]) technique by David Blei and his collaborators to the point where at the 2017 Text-as-Data conference at Princeton,[12] virtually every paper presented employed some form of topic modeling. LDA has been widely implemented in the two now-dominant data analytics platforms, *R* and Python, it is reasonably computationally efficient even on fairly large data sets (a distinct contrast to methods such as word embeddings which are based on neural networks) and it's [many] quirks are relatively well understood. The output of an LDA analysis will be a series of themes based on sets of words (which *usually* will make the core concepts of the theme fairly evident); the texts in the corpus can then each be assigned a probability of belonging to one or more—or none—of these themes.

Taking the advantages of the approach as a given, the quirks of concern to us here will be:

- As a method dependent on numerical optimization in a high dimensional space characterized by many local maxima—which is, of course, typical of many if not most text analysis approaches beyond the most simple—in most real-world problems, multiple estimates of the themes and to some degree the classification of texts into themes will be different with each run of the program (presuming the random number generators are initialized from different seeds)
- As a "bag of words" method which [typically] does not include explicit semantic or grammatical information, LDA models have a tendency to produce some nonsensical "themes" (and classifications) which, while evident in retrospect and definitely in the data, are of no use to an analyst.[13]

[11] Inconveniently the same acronym as the vaguely related "linear discriminant analysis" and I watched at least one DARPA-sponsored workshop where the acronym was simultaneously used for both methods.

[12] Coincidentally, presumably, the birthplace of LDA

[13] An interesting example I saw of this once was an analysis of Congressional foreign policy debates which produced one thematic cluster based on a series essentially nonsensical terms. The computer scientist who had done the analysis could make no sense of these whatsoever and assumed there was some bug in the program; the political scientists immediately recognized these as the code names for a variety of U.S. military operations which, of course, had been selected precisely so they did not convey information.

4 Data preparation

The corpora used in this exercise involved about fifteen months of news articles obtained from a combination of Factiva and the U.S. government Open Source Enterprise.[14] These primarily dealt with the Middle East, and were an combination of articles in English and articles in Arabic which had been machine-translated into English.

4.1 Pre-filtering using an event coder

The initial filtering step was to select only sentences which had produced events using an automated event coder which is apparently a slightly updated version of that used to produce the ICEWS data (`http://thedata.harvard.edu/dvn/dv/icews`). This coding was done by another contractor on the project and the coder is proprietary, but my understanding is that the event-resolution engine is essentially that used for ICEWS except that the actor dictionaries are very substantially larger, by a factor of two or more, than the ICEWS dictionaries, and I wouldn't be surprised if many of these enhancements focused on the Middle East. The clear advantage of using event coding as a pre-filter is that the corpus now consists only of sentences which involve at least some (or at least some coded) political interaction.

[If one is trying to replicate—or more likely, just use—this approach, my guess is that any of the available open source coders—TABARI, PETRARCH-1, PETRARCH-2, Universal-PETRARCH[15] will produce fairly similar results based on the arguments I've made in more detail here:

`https://asecondmouse.wordpress.com/2017/02/20/seven-conjectures-on-the-state-of-event-data/`

I would particularly note the first finding of that essay—illustrated in Figure 1—that the marginal distribution of events produced by the proprietary ICEWS coder compared to the two open-source PETRARCH coders does not differ dramatically except in a couple of categories, with the [Dept of Defense-funded] ICEWS coder picking up more events in the CAMEO 18x "unconventional violence" category and PETRARCH-2 picking up excessive events in the CAMEO 04x "meeting" category due to an easily-corrected mistake in the dictionaries. The alignment of these marginal distributions does not preclude the *possibility* that the ICEWS coder is in fact accurately detecting, say, twice as many interactions as the open-source coders, but this seems fairly unlikely.[16] The open source coders are also likely

[14] Previously known as Open Source Center, World News Connection, and, for a very long period of time, ca. 1941-2000, Foreign Broadcast Information Service.

[15] Open source code and dictionaries for each member of the PETRARCH "family"—they are in fact three *very* different programs, though all in Python—is available at `https://github.com/openeventdata.` Code for the C++ TABARI is available at `http://eventdata.parusanalytics.com/software.dir/tabari.html` or `https://github.com/philip-schrodt/TABARI-Code`

[16] In the absence of access to the ICEWS source texts, it is impossible to ascertain whether this is the case. The original ICEWS coder is nominally available for license under onerous restrictions, possibly but not necessarily involving the sacrifice of first-born children, for academic and not-for-profit research, though I've yet to hear of anyone who has actually done this, but that would also allow this to be tested on, say, the

to have reasonably comparable performance on identifying state-level actors—the focus of this analysis—even if the much larger dictionaries of the ICEWS-derived coder are almost certainly better for sub-state actors.]

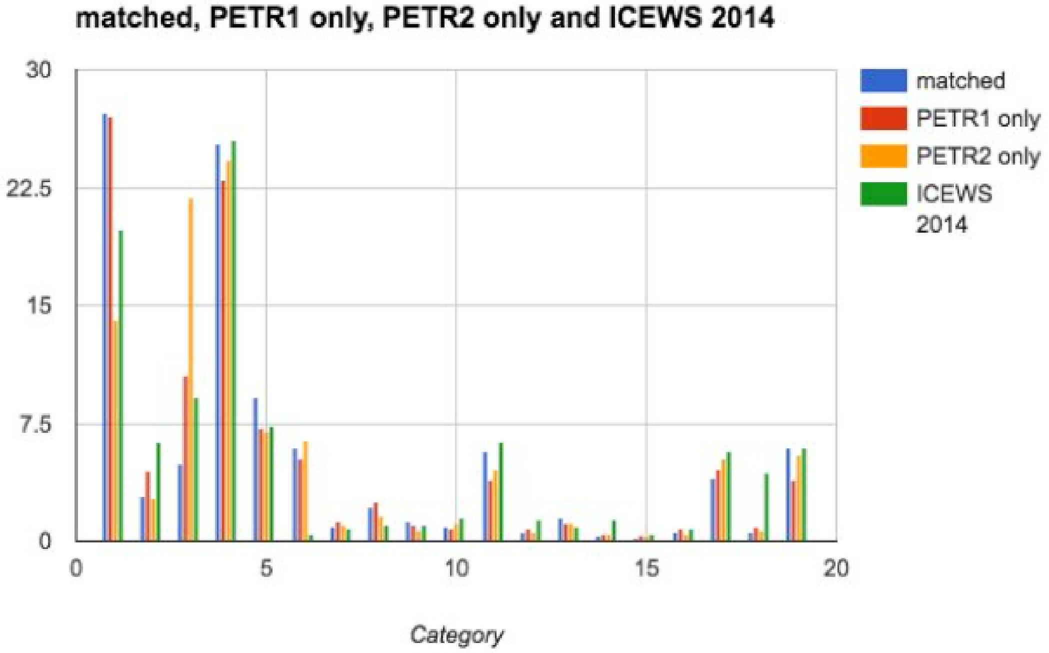

Figure 1: Marginal distribution of CAMEO cue categories in events generated by three distinct automated coding programs

In the nation-specific analysis, the event pre-filtering is followed by a string filtering for the name (and synonyms for the name) of the state being analyzed. These can occur anywhere in the sentence, not just as a source or target in the coded event so, for example, if the coded event involved the US and Saudi Arabia discussing Qatar (hence the event actors are USA and SAU), that sentence would still be included.

4.2 Stop word elimination

A series of analyses by Spirling and his collaborators[17] have demonstrated that routine preprocessing methods such as stopword, punctuation and number removal, stemming, and removal of low-frequency words can have important effects on the clusters produced by an LDA analysis. While this work has mostly been done on texts such as treaties and campaign

Gigaword corpus. Though given the rapid development of newer approaches to event coding, both dictionary-based and example-based (e.g. through neural networks) such an exercise is probably not worth the trouble as the ICEWS coder is based on ca. 2010 technology and the constituency-parse-based PETRARCH-1/2 are being superceded by open-source coders based on the newer dependency-parsing approach.

[17] From PolMeth XXXIV: `https://polmeth.polisci.wisc.edu/Papers/DennySpirling2017.pdf`

materials that are likely to be quite different than sentences from news articles, an analysis of the *New York Times* showed stop-word removal had some significant effects. With this in mind, a relatively limited set of stopwords based on an examination of the frequency of words in the texts themselves, rather than the use of a general-purpose stopword list such as that available in `gensim.parsing.preprocessing.remove_stopwords`, was used, with a small amount of iterative addition of stopwords based on the appearance of common words in the original estimates of the clusters.

In the country-specific analyses, the names (and synonyms) of the target country were removed, since these were occurring in every sentence in the text corpus. Words are also lowercased (though not stemmed) and both punctuation and any string containing one or more digits are removed.

4.3 Synonym resolution

A customized set of synonym sets shown below—the first string in the list is the standardized string which is substituted for any of the remaining strings in the list—is used to standardize both multi-word entities such as `United States, Saudi Arabia` and `United Nations`, resolve demonyms (`American, Yemeni, Qatari`), and deal with other common idioms such as the use of a capital city (`Washington, Riyadh`) to refer to a government. This list was largely developed incrementally on the basis of synonyms which appeared in the thematic lists, but in a more generalized system the `CountryInfo` file[18] could be incorporated into the system: this is a comprehensive list of country names, synonyms, demonyms and other alternative forms, major city and region names, and national leaders for about 240 countries and administrative units. In this application, I did not resolve leader names such as `Trump` and `Assad` to the country name, though in other applications that might be appropriate.

```
synsets = [[" saudi_arabia ", " saudi arabia ", " saudi ", " riyadh ", "kingdom of saudi arabia "],
         [" united_states ", " united states of america ", " united states ", " usa ", " america ",
         " american ", " washington "],
         [" united_nations ", " united nations ", " secretary general ", " un ", " security council "],
         [" abu_dhabi ", " abu dhabi "],
         [" egypt ", " egyptian ", " cairo "],
         [" yemen ", " yemeni "],
         [" iran ", " iranian ", " teheran "],
         [" gcc ", " gulf cooperation council "],
         [" uae ", " united arab emirates "],
         [" gaza_strip ", " gaza strip ", " gaza city "," gaza ", " rafah "]]
```

4.4 Short sentence and low-frequency word removal

The system requires that “sentences” contain at least 256 characters to be included in the analysis. This restriction is done both to eliminate strings that aren’t really sentences such as headlines, sports scores and weather reports (the filter on the source texts left something

[18] https://github.com/openeventdata/CountryInfo

to be desired in this regard. . .) and to increase the likelihood that there would be a sufficient number of words in the sentence that it could be reliably classified. The 256-character limit is, of course, arbitrary (and is set as a global in the filtering program) but based on earlier work I've done on event data, where sentences in wire service reports which describe codeable political interactions almost always exceed this length. As discussed below, in this particular application, the major issue turned out of be sentences translated from Arabic that were too long, not sentences that were too short.

Words which occurred in fewer than 5 times in the first 2048 cases were removed: as with everything in natural language processing, the distribution of words had a very long tail and in fact I'm guessing that this $N \geq 5$ is probably lower than necessary.

A processing step that may have been problematic involved building the dictionaries only on the first 2048 records, which was done because I was worried about the total memory and processing time requirements, which turned out not to be an issue. In the country-specific cases, I was analyzing only a month of data at a time, and the records were not in chronological order—in the data set I was working from, produced by another contractor, they were ordered more or less randomly—so this worked well. In the larger set of data, I think there was some chronological structuring and a better approach would have been to randomly sample, and probably from a larger set of records: I doubled the sample (and the minimum word threshold) for this but a substantially larger set probably should have been used and would have been quite feasible.

5 Magic sauce: merging multiple models

The LDA models were estimated using Radim Rehurek's Python[19] open-source `gensim`[20] system, specifically the routine `models.LdaModel()` with the default hyperparameters, which was used first to estimate a model and then saving, for the downstream processing discussed below, the sentences which are classified to any topic with a probability of 0.50 or higher—this threshold is an arbitrarily-set hyperparameter, of course—to files along with their bag-of-words vectors and assorted other information. As noted earlier, in large, real-world corpora such as those being analyzed here, as distinct from "toy" problems, LDA does not produce a unique set of results because of the presence of local maxima[21] in the optimization surface. Four models with either eight (country-specific) or sixteen (global) topics were estimated in each experiment; these typically required about five minutes to run.

I promised earlier—remember, this is an application-oriented paper, not something designed to eventually impress a dean, associate dean, assistant dean, deanlet, deanling or, more

[19] For what it's worth, for practical reasons relating to my limited cognitive bandwidth, I've shifted all of my data analysis to Python, abandoning *R*, since I've yet to find anything I'm not able to do in Python and, of course, Python is a far more suitable environment when working with text, which is what I spend most of my time doing. And, okay, confession: I never really learned to think in *R*, whereas I've no difficulty thinking in Python (or C).

[20] `https://radimrehurek.com/gensim/`

[21] Or minima, depending on how you are framing the optimization problem.

lucratively, search committee—to reflect on whether the software is worth investing the effort apply it: hey, `gensim` is cool! The software worked right "off the web,"[22] requires very little infrastructure code (my "glue" programs were only a few hundred lines, and invoking the core estimation procedures required just about a dozen lines), and runs quite quickly even on a relatively upper-end—if aging—desktop.[23]

A program called `topic_formatter.py` is the chronology-generating workhorse, integrating the results from LDA-estimation step using the following approach:[24]

1. Align the documents to topics—the extracted topics not only vary across the runs of the topic model, but their ordering is arbitrary—which are more or less the same across the multiple estimates by first finding the pattern which is most common (for example, a document might be assigned as cluster 3 in the first run, 5 in the second, and 2, 7 in the last two, so its pattern is [3, 5, 2, 7]: this is the seed for a "consolidated cluster". Assign additional sentences to the consolidated cluster if they agree on any three out of four of these assignment, e.g. a sentence with the assignment [3, 5, 3, 7] would qualify; [3, 4, 2, 7] would not. Keep track of the total probabilities of the topic terms.

2. Combine any consolidated clusters where the correlation between the terms in the consolidated cluster is greater than 0.95.[25] These are the "super-clusters."

3. Then for each supercluster:

a. Get all of the unique sentences and sort by date.

b. Use the `gensim summarize()` function to get "representative sentences" subject to hyperparameters on the

maximum length (characters) of a representative sentence : some of the Arabic sentences get really long, particularly those using customary invocations of the many virtues of assorted rulers who just happened to be controlling the news source, and don't make for very good summaries. This maximum is currently set at 512 characters.

Total length of the set of representative sentences: The speed of the `summarize()` function decreases exponentially (or thereabouts) with the increasing size of the text being summarized, so this is currently limited to a total of 32,768 characters,[26] which is probably too conservative.

[22] Formerly "out of the box" but software no longer comes in boxes, and even better, it's "free as in puppy"!

[23] late 2013 iMac with a 3.2 Ghz Intel Core i5 and 16 Gb of memory, though I wasn't pushing the memory capacity at all. Rehurek has done extensive work on configuring the system so that it can scale to work with datasets too large of fit in memory, though that wasn't needed here.

[24] If you really want to see how this is done, I'd strongly advise just asking me for the code rather than trying to figure out what I've done from the English-language description below, which is rather like viewing the code through one of those shimmering SFX sequences invoking an alternative universe where emotionally troubled individuals with remarkable powers spend most of their time harmlessly throwing very large objects at one another, while crushing cars in our universe. But I digress: just email me to get the code.

[25] A hyperparameter called `CLUST_THRES` in the code: this quite high threshold seems to work well.

[26] As is evident in this discussion, as an old C programmer I like powers of 2, but that's just me: I doubt those choices have any discernible effect on code implemented in contemporary scripted languages such as Python. But it can't hurt.

Curiously, the `summarize()` function—which seems to be a work in progress—does not always return results, so when this fails we just use the first three sentences in the chronology as the representatives.

c. For each day, eliminate duplicates by comparing the common membership $C(n, m)$ between the bag-of-words (BOW) vectors for each pair of sentences—that is, the proportion of the words in sentence n that are also found in sentence m—using the following rules:

symmetric elimination of the second text (the sentence order is presumably more or less random as usual) when the common membership of the BOW for sentences $C(n, m)$ and $C(m, n)$ are both $> DUP\_THRES$ (currently set at 0.90): here both texts are essentially the same

subset elimination on [n] when $C(n, m) > DUP\_THRES$ and $C(m, n) < DUP\_THRES$: here n is a subset of m and we retain the longer text

Note that because we are comparing on BOW counts following stopword elimination and synonym resolution, this approach is probably closer to evaluating on topic similarity than string matching. The use of the $C(n, m)$ metric to measure sentence similarity is only one of any number of possible metrics that could be used; note also that because $C(n, m)$ is a proportion based on the number of words in n, $C(n, m) \neq C(m, n)$ unless the two sentences contain the same number of words.

[The most common source of near-duplicate sentences, by the way, are cases where the machine translation of the identical Arabic texts—typically from a common Arabic-language wire service story—into English has resulted in slightly different synonyms being used in a couple of places in the sentence. The translations were handled by another contractor and I don't have access to the full original Arabic texts, so I don't know whether this is due to different translation programs being used, or some of the translations being obtained from the `/en/` branch of the source's web site if this is available, or due to come context-dependent word choice being affected by differences earlier in the story, but it wouldn't surprise me if we see this near-duplicate issue more frequently as machine translation is used more as a pre-processor.]

6 Results

The output of the formatter are the super-clusters with a header showing the keywords, the numbers of the original clusters (which are themselves irrelevant but indicate how frequently each theme was found), some representative sentences, and a summary of the events, source actors and target actors (which we haven't done much with):[27] an example is provided below. This is followed by a chronological listing of the events in the single-country example, or the frequency of events in the general example.

[27] As evident from the example, the event coder being used has a fairly expansive concept of "actors," including [lots of] pronouns as well as extended noun phrases.

```
Chronology for theme {2, 3, 5, 6, 9, 10, 11, 12}
Keywords: arab, candidate, organization, against, terrorism, united_states, minister,
          international
Representative sentences:
 -- The foreign Minister Sameh Shukri, had announced Egypt's support for the candidate of the
    French Azoulai in the last round in the elections of the new director general of UNESCO
    during the final round of the elections that took place yesterday evening against the Qatari
    candidate Hamad al-Kuwari, after having lost the candidate of the Egyptian tour of the
    return address before the French candidate Oder, where Egypt's candidate took place on 25
    votes against 31 candidate obtained France.
 -- Paris: Qatar issued a candidate, France, the race for the leadership of the United Nations
    Educational, Scientific and Cultural Organization "UNESCO" after a third round of voting on
    Wednesday, limited the number of competitors over its chairmanship five, while accused the
    Egyptian Foreign Minister Sameh Shukry in an interview with "Egypt today" newspaper, Qatar,
    "The use of its financial authority to influence the executive council of UNESCO, which
    includes 58 members.".
Most common event elements:
  Cue       Event        Source actor                         Target actor
 80 05     78 051       9 He                                14 Egypt
 45 01     45 010       7 Egypt                              8 Qatar
 15 11     15 190       4 who                                8 France
 15 19     11 111       3 his                                4 the Arabs
 12 02      9 020       3 that                               3 them
  8 09      7 141       2 its                                3 its candidate
  7 14      6 090       2 which                              3 the African group, where th...
  5 10      4 112       2 his country                        2 its
```

The system has been tested on two country-specific corpora—Qatar and Yemen for October 2017—and a much large corpus of about 2-million sentences (reduced to about 750,000 following the event filtering) generally covering the Middle East. The 16 topics with the highest number of sentences assigned to them were retained for the single-country case, and 32 topic were retained for the general case. The eight most highly weighted keywords for the themes for the Qatar case are shown in Appendix 1, and for the general case in Appendix 2. These keywords, of course, give only a general sense of the themes (the full chronologies, leaving aside intellectual property restrictions on distribution of the texts, are hundreds of kilobytes for the single-country-month cases and tens of megabytes for the general case) so in the spirit of "trust me…" here's my read on the results:

6.1 Stuff that works

The themes generally make sense: Overall, the system works fairly well in the sense of extracting and differentiating the multiple streams of political interactions that the country is involved in. For the country-month case, this tends to be organized—as a human would be likely to organize—by the state or organizations the state is interacting with; in the general case the themes seem to be more behavioral. Major themes that one would expect to see—notably violent conflict in Yemen in the Yemen-October-17 run and the Syrian civil war in the general case—come through very clearly, but so do more focused themes such as the Arab reaction to the US opening an embassy in West Jerusalem. The dominant "UNESCO election" theme illustrated above was unexpected as this controversy got very little coverage in the Western press, but was quite a kerfuffle in the Arab world and in fact was clearly important in October 2017.

The sentence assignment also usually makes sense but not always: In addition to finding credible themes, the chronologies are also *usually* successful in putting together interactions that a human analyst would consider belonging together. But not always: there are definitely some head-scratchers of "why the heck is that in there?" which presumably are due to chance combinations of a large number of words with relatively low topic classification weights which are jointly sufficient to move the sentence past the threshold for inclusion in the cluster. Since we have yet to get feedback from analysts involved in day-to-day operations on the clustering, we don't know whether they will be willing to just ignore these or whether we will need to tighten the inclusion criteria.

Multiple runs find the same core clusters: The multiple-run approach is quite good for ascertaining the relative importance of clusters, as the core behaviors are not only found consistently across multiple runs but often occur as multiple themes within a single run, presumably as a consequence—possibly due to different sources and/or the native vs. translated English differences, though I've not checked for this—of the same general behaviors being described with consistently different sets of synonyms.[28]

Less important clusters are not found in all runs: As we had hoped, the multiple-run approach also yields some quite substantively meaningful clusters that are found only in a subset of the runs: in the single-country case these clusters tend again to focus on interactions with a specific other actor. That said, some of the clusters found only in subsets of the run do not seem very meaningful: one possibility I've not experimented with is increasing the number of runs and then setting a threshold for the number of instances of a theme before it is included.

The system has modest computational requirements: The system runs reasonably fast, with the entire analysis from the original corpus to the chronology taking about ten minutes for the country-month case and about half an hour for the general case.[29] This is not sufficiently fast for real-time human interaction, but is still within a range where global monitoring could be done with resources on the order of cloud computing and not super-computing.

6.2 Stuff that didn't work so well

Many themes in the general run are too generic: As can be seen from Appendix 2 and as discussed above, the general analysis picks up some expected major themes such as the Syrian civil war, the US opening an embassy in West Jerusalem, protests

[28] I experimented with looking for "sub-themes" by relaxing the inclusion probability and then running topics just on the sentences of a super-cluster: this did not produce anything substantively meaningful that I could discern, and mostly just random clusterings that I'm assuming are due to synonym use.

[29] In fact, it seems a bit *too* fast for the general case, and as noted below there might be something a bit off with the corpus.

in Turkey and Egypt, and probably some general themes on economics, education and local politics.[30] But a lot of the keyword lists seem very vague and mostly just permutations of what are apparently common words. There are at least three possible causes for this:

- Those common words need to be added to the stopword list
- The criterion for combining clusters may need to be loosened to handle the vastly larger amounts of text in the general case: presently very few (like "none" in the example here) of the individual thematic clusters are being combined
- Theoretically very uninteresting, but there may be some screwy duplication going on in the corpus itself: we've only recently noticed this and are still looking into that possibility

The `gensim summarize()` function fails quite frequently: It does, and I haven't explored the reasons why in any detail. Automated summarization is a rapidly evolving field in its own right and the approach incorporated into `gensim` may not be the best available.

It's difficult to assign sentences to themes outside a country-month sample: One thing we'd like to be able to is find precursor and successor sequences to themes that are dominant in a given month, and connect these across months. In other words, a theme that is clear in one one month may not be sufficiently prevalent in earlier or later months—unlike, say violence in Syria or Yemen, which is a clear theme in all months—but individual sentences might still be identifiable. Thus far efforts to do this haven't been very successful but I may be doing something wrong in reconciling the `gensim` BOW vectors across months.

The heterogeneity of sentence length is problematic: This became evident in another exercise where I was trying to distinguish between sentences in the middle of a thematic cluster from those on the edges: attempts so visualize this showed that the total number of words was far and away the dominant dimension, and as noted above, there are major stylistic differences in the native-English vs translated-Arabic texts in this regard. Alternative clustering metrics may be able to deal with this.

7 Final remarks

So, have we found that mythical beast, the chronology generator, the core tool of the equally mythical political analyst's workstation? The mythical system that will change the qualitative analyst's attitude from "Leave me alone" to "I gotta have one of those!"

[30] I'm being vague here as the chronologies in the general analysis are so huge that it is not really practical to read them, unlike the country-month chronologies: we're working on figuring out how to summarize these more usefully.

Well, not yet, but these results appear to be moving in a credible direction: maybe we are even beginning to see shimmering outlines of the mythical creature through the fog? I see at least four advances here:

- Using the event prefilter to restrict the corpus to sentences which generate codeable political events insures that the corpus contains mostly the sorts of transactional political activity which analysts expect to see as the primary components of an historical chronology.
- While it clearly can be refined further, the approach of generating super-clusters from multiple runs is generally producing credible thematic clusters, as well as providing a rough metric as to the prevalence of each theme.
- The system runs on inexpensive conventional hardware with most of the heavy lifting done with a single open source program, and the processing time is in minutes, so it is possible to experiment with relative ease. In contrast to many earlier efforts to create the mythical analyst's workstation, there are no black-boxed proprietary components.
- Except for a few of the pre-processing steps (e.g. synonym sets and stopword lists, both fairly general) the system is entirely data-driven—that is, unsupervised—and does not depend on human-generated ontologies or exemplars. This in particular means it will be trivially easy to apply in specialized domains provided a filter with equivalent functionality to the ICEWS event coder is available.

This is an on-going (if, at the moment, relatively low-level) project and further work is likely to focus on the following

- There are perhaps twenty major hyperparameters at various steps of the process, many of which will affect the tradeoff between precision and recall, and I've only begun to experiment with these.
- In keeping with the overall "taming the firehose" theme, summarization is probably critical to this exercise, both with respect to the themes, and also within individual days, particularly when these can potentially include a very large number of events (e.g. looking at Syria, Israel, or Egypt rather than Qatar and Yemen). As noted above, exploring alternative open-source possibilities for this could be useful.
- The current measures of the centrality of sentences to the thematic clusters aren't working very well, and getting some method—correspondence analysis?—to allow the clusters to be visualized in two or three dimensions would be useful, particularly if there are identifiable subclusters.
- Some optional form of setting priors on both the themes and their contents would probably be useful from an analysts perspective: for example optionally overweighting specific actors (states, leaders or organizations) and providing a set of examples for what the analyst wants to see in a theme, and then building a theme around it. There are variants on LDA which place greater emphasis on its Bayesian elements, or some other method, such as metrics for similarity detection that work better than those we are currently experimenting with, might be appropriate.

8 Appendix 1: Themes in a state-specific chronology

```
Theme {2, 3, 5, 6, 9, 10, 11, 12}: arab, candidate, organization, against, terrorism, united_states,
minister, international

Theme {0, 1}: united_states, against, president, state, minister, crisis, iran

Theme {4, 15}: foreign, minister, iran, arab, terrorism, government

Theme {7}: united_states, state, minister, crisis, terrorism

Theme {8}: united_states, minister, state, president, arab, foreign

Theme {13}: state, against, international, united_states, terrorism

Theme {14}: united_states, state, against, arab, organization
```

9 Appendix 2: Themes in a general chronology

```
Theme {0}: forces, syrian, turkish, military, army, killed, syria, against
Theme {2}: syrian, turkish, forces, military, syria, army, turkey, killed, civilians

Theme {17}: united_states, palestinian, israeli, israel, president, jerusalem, trump, occupation,
          gaza_strip
Theme {23}: palestinian, israeli, united_states, israel, jerusalem, occupation, gaza_strip,
          palestinians, president

Theme {18}: united_states, president, united_nations, against, trump, egypt
Theme {30}: united_states, against, government, president, education, protests
Theme {21}: president, united_states, egypt, elections, presidential, party

Theme {7}: against, united_states, government, party, turkey, protests
Theme {20}: against, party, government, turkey, protests, protest

Theme {5}: company, ministry, companies, trade, sector, education, government, egypt, investment
Theme {28}: company, ministry, companies, trade, education, egypt, students, party, sector

Theme {1}: education, students, party, city, council

Theme {3}: support, ministry, development, work, education, students, party, city
Theme {14}: ministry, support, company, development, work, education, government, students, party

Theme {4}: team, club, egypt, player, league, alahli, players, technical, football

Theme {6}: city, forces, killed, journalists, ministry

Theme {8}: international, support, party, egypt, city

Theme {9}: investigation, court, education, students, party, accused, city, council
```

Theme {10}: minister, ministry, council, elections, law, committee, government, parliament, presidential

Theme {11}: support, international, party, egypt, efforts, work, city, council

Theme {12}: court, accused, police, investigation, arrested, case, ago, few

Theme {13}: law, government, minister, council, committee, ministry, parliament, elections, approved

Theme {15}: students, education, university, team, school, party, city, club

Theme {16}: international, support, security, countries, organization, egypt

Theme {19}: general, director, city, journalists, ministry

Theme {22}: university, women, party, few, city

Theme {24}: security, against, party, city, council

Theme {25}: egypt, international, city, media, journalists

Theme {26}: against, ago, few, city, journalists

Theme {27}: egypt, city, media, journalists, ministry

Theme {29}: saudi_arabia, minister, sheikh, visit, prince, abdullah, king, health, president

Theme {31}: support, security, international, efforts, countries, organization, egypt, terrorist, terrorism