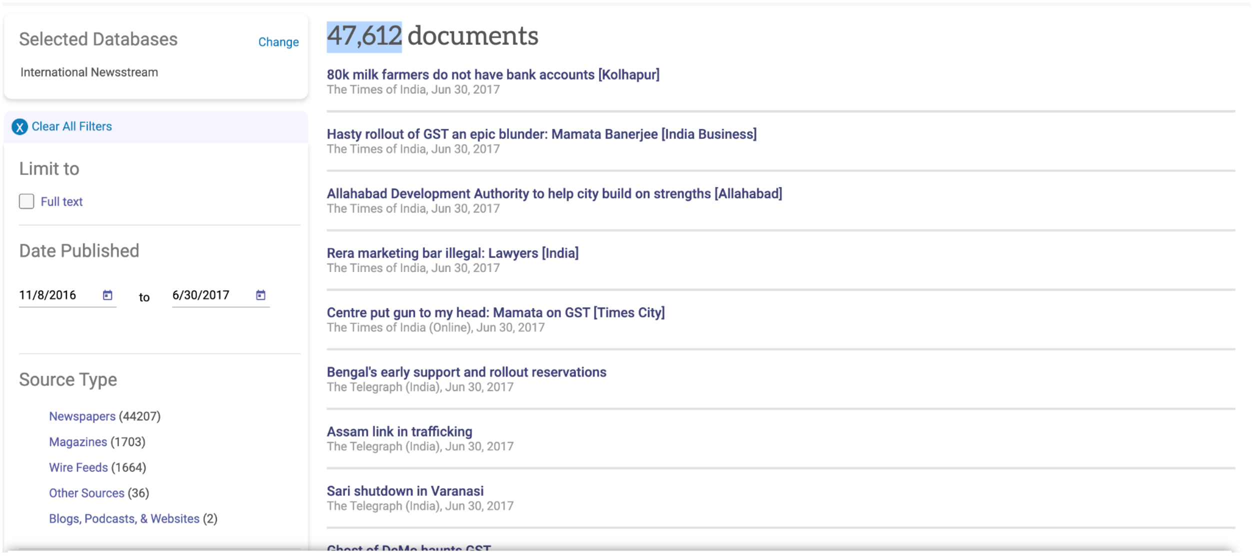
Data set creation:

Filter step	Query	# of documents
Use ProQuest's International Newsstream database	N/A	204,769,178
Published in India	PBLOC(india)	10,848,595
Published in India and published between when demonetization was enacted and the end of June the next year	PBLOC(india), 11/8/2016 - 6/30/2017	858,310
Published about demonetization	demoneti?ation	105,908
Published in India, about demonetization	PBLOC(india) demoneti?ation	74,982
Published in India, about demonetization, in the time frame	PBLOC(india) demoneti?ation, 11/8/2016 - 6/30/2017	42,022
Published in India, about demonetization, in the time frame, including any word form of "demonetization"	PBLOC(india) demoneti?ation OR demoneti?e? OR demoneti?ing	47,612



Selected Databases Change

International Newsstream

Clear All Filters

Limit to

□ Full text

Date Published

11/8/2016 to 6/30/2017

Source Type

Newspapers (44207)

Magazines (1703)

Wire Feeds (1664)

Other Sources (36)

Blogs, Podcasts, & Websites (2)

47,612 documents

80k milk farmers do not have bank accounts [Kolhapur]
The Times of India, Jun 30, 2017

Hasty rollout of GST an epic blunder: Mamata Banerjee [India Business]
The Times of India, Jun 30, 2017

Allahabad Development Authority to help city build on strengths [Allahabad]
The Times of India, Jun 30, 2017

Rera marketing bar illegal: Lawyers [India]
The Times of India, Jun 30, 2017

Centre put gun to my head: Mamata on GST [Times City]
The Times of India (Online), Jun 30, 2017

Bengal's early support and rollout reservations
The Telegraph (India), Jun 30, 2017

Assam link in trafficking
The Telegraph (India), Jun 30, 2017

Sari shutdown in Varanasi
The Telegraph (India), Jun 30, 2017

Ghost of DeMo haunts GST

Textual analysis:

Overview: We want to analyze each document in our dataset to see if it meets our criteria for potentially mentioning demonetization happening again. We do this by extracting the text from each XML document (which is how ProQuest stores the articles for analysis) and parsing it for any of many possible bigrams (two-word combinations) that could indicate they are relevant to our question. An example bigram could be “happen again” to indicate that the author of the article believes that the government may do another demonetization. If the bigram is found in the text of the article, then we search the 10 words before and the 10 words after the bigram to see if we can also identify the word “demonetization” or a synonym of “demonetization.” Repeat this for all combinations of bigrams and synonyms, keeping track of the 20-word snippets each time we find both a bigram and a synonym close to each other. This methodology is based on several papers that have previous explored simple sentiment analysis: Hassan et. al., and Bloom et. al., with the main idea being that bigrams are very specific phrases that usually only have one meaning, so we then assume that if demonetization is mentioned nearby, it may be in connection to the meaning of “happening again.” If we determine if an article has a bigram and a synonym close to each other, we save the article as a row in a CSV file, recording the title, author, date, location if possible, publisher’s location, and any snippets we found that represent a bigram in proximity to a synonym.

Implementation:
First, write the relevant row headers to the CSV file at the path defined in the code (overwriting whatever was there before, so be careful). Iterate through each document in the subset we are

examining currently. Extract the XML and text using lxml and BeautifulSoup. Pass just the text of each article to a function that parses it for matches.

The parsing works as follows: create two arrays of words, one with punctuation removed, one without. For each bigram of interest that we have defined (e.g. “happen again”), check each word in the punctuation-removed array to see if it matches the first word of the bigram. If it does, check if the next word matches the second word. If it does, take the subarray (being mindful of the boundaries of the larger array) that is plus/minus *n* words around where we found the first bigram, where *n* is defined by a constant elsewhere in the code. For each synonym that we have defined (e.g. “the event”), check if it exists in the subarray of words we just created, in a similar manner to how we checked for bigrams. If we find a match, then take the same subarray from the larger array of words that we didn’t remove punctuation from (plus two words on either side in case there are offset errors created by punctuation), and join the words into a string. This string is now a relevant snippet of an article’s text. Add it to the array of relevant snippets we’re maintaining for this article. Once all combinations of bigrams and synonyms have been checked for, return all snippets found.

Back in the main iteration function, if the list of snippets found by the parsing function is not empty, then we need to extract information about the article itself to save to our CSV file. Leverage XML attributes to get the date, title, source, and location of the article. In particular, obtaining the location works by first checking if the article is from certain sources that have known ways to store the location (e.g. The Hindu keeps its location information in an XML attribute called “DocLocation” that most other articles don’t have). Otherwise, we guess that some articles have the format CITY, DATE: Rest of article text, or CITY: Rest of article text, so use that to try and get some sensible information. Write all the meta information to a single row in the CSV file. Then, proceed to processing the next article.

Bigrams:
The following is a list of two-word combinations that indicate that an article is potentially relevant to our research question. If the two words appear adjacent to each other in this order, we search the nearby words for any synonym of demonetization.

[might again
could happen
happen again
do again
it again
never happen
never again
government may

RBI may
likely to
unlikely to
likely never
likely will
possible that
once again
another round
new currency
new notes
2,000 new
2000 new
thousand new]

Synonyms:
The following is a list of one or two-word phrases that have similar connotations to "demonetization," in some word form (verb, gerund, noun, etc). If we detect any of these words/phrases in proximity to a bigram, we assume that the sentiment is relevant to our question of demonetization happening again.

[the event
the policy
the cash ban
the move
the announcement
the initiative
the ban
demonetize
demonetizing
demonetise
demonetising
ban]

Results:
We produced a CSV file recording the title, source, date, location, publisher location, and the relevant excerpt(s) of each article that is deemed to possibly help answer the research question. The file has 1,183 articles, one per row.

Full article information:
We take all 47,612 articles in our dataset (as a refresher, obtained by filtering for published in India, the specific date range, and some word form of “demonetization”). For each one, save the title, source, date, and location if it can be parsed to a CSV file for further analysis and benchmarking.

Google Trends replication:
We create 6 different datasets, one capturing the sentiment of each Google Trends search as described in Appendix E.1 of the original paper. The dataset consists of the ProQuest International Newsstream database, filtered to include only articles published in India from the beginning of October 2016 to the end of June 2017.

Most of the 6 datasets are extremely small as compared to the main demonetization dataset (containing hundreds of articles as opposed to tens of thousands). We try to capture the sentiment of each Google Trends search by including many synonyms for the term, e.g. the dataset for “2000 Rs” includes searches for “two thousand rupees,” “Rs 2000,” and more.

Here are the search parameters for each term:
2000 Rs: PBLOC(india) ("2000 Rs" OR "two thousand rupee" OR "Rs 2000" OR "Rs 2,000" OR "2000-rupee")
ATM Line: PBLOC(india) ("ATM line" OR "ATM lines" OR "ATM queue" OR "ATM queues" OR ATM NEAR/3 line OR ATM NEAR/3 wait)
ATM Cash Withdrawal Limit Today: PBLOC(india) ("withdrawal limit" OR "withdrawal limits" OR "withdrawal cap" OR "withdrawal caps" OR "withdrawal max" OR "withdrawal maximum")
Cash: PBLOC(india) cash
ATM Near Me With Cash: PBLOC(india) (ATM NEAR/3 "with cash" OR "ATM near you" OR ATM NEAR/3 "has cash" OR ATM NEAR/3 "with money" OR ATM NEAR/2 cash)
Authority Letter For Cash Deposit:
PBLOC(india) ("authority letter" NEAR/5 (cash OR deposit OR bank) OR "authorization letter" OR "authorisation letter" OR "authority letter for cash deposit" OR "bank authorization" OR "bank authorisation" OR "authorized person" NEAR/3 (bank OR deposit) OR "authorised person" NEAR/3 (bank OR deposit) OR "authority letter" OR "authorising in writing")

We produce a CSV file with 62,470 articles total, and with columns for the location, publisher location, title, source, date, and which search term the article corresponds to (0-5, where 0 is the search for “2000 Rs” and 5 is the search for “Authority Letter For Cash Deposit”).