Methodology, Pipeline, and Results: Details

Figure 1: A flowchart of the computational steps executed in our end-to-end pipeline. The salient computational steps are described in detail below and in less detail in the methodology section of the paper.

Code for this project and instructions for running the code are available through a github repository. Access can be requested by contacting the corresponding author.
https://github.com/Roychowdhury-group/FENESTRA-Fake-News-Structure-and-Threat-Assessment

Each block mentioned in Figure 1 diagram is described below:

**Block 1**: Data is generated from various sources, including key statistics for the two datasets
Bridgegate:
- Number of documents: 385
- Number of sentences: 20433
- Number of dependency tree based relationships: 21667
Number of srl-based (A0,V,A1) relationships: 25693 (58% of the total SRL extractions)
Number of SRL extractions in total: 44212

Pizzagate:
Number of posts: 17,948
Number of sentences: 42979 (29790 of them has a length of greater than 30)

Block 2: Preprocessing step. We use the resources from the previous block in order to obtain noiseless, relevant and informative raw text data as pipeline input.

Tools:
- warctools
- pandas
- boilerpipe
- guess_language
- adblockparser
- Beautifulsoup

Some examples of the preprocessing steps include:
- warcindex to find all HTTP responses
- warcpayload to extract pages possibly containing useful texts
- pandas to remove duplicate URLs
- removal of spam web pages

Block 3: One of the outputs from the previous step are date-time stamps. In the analysis of conspiracies and conspiracy theories, it is helpful to know the timing of events. The plot below shows how timing information can be used for tracking the emergence of different actants and events associated with them.

Blocks 4,5,6: Two types of output are produced in this step. First, we apply paragraph-based coreference resolution where each post is treated as a paragraph. Second, we parse each sentence individually to extract the relations, using nltk to segment paragraphs into sentences.
Block 7: Using the coref part of Stanford corenlp package, the paragraphs/posts are used as input with each input to the coref resolver comprising several sentences. The number of sentences varies given the disparate lengths of posts. The output of coreference is translated into a mapping dictionary, where each word (usually a pronoun) in a sentence is mapped to another word (ideally a non-pronoun noun). We index the corpus so that each word in paragraph has a unique sentence number and unique word ID, often corresponding to its sentence index. The mapping dictionary, as a result, is from word ID to word ID. An example of the coreference output is shown below:

Block 8: We used the sentences from the previous step and extracted the dependency trees using the stanford corenlp package.

Block 9: In this step, we use the output of the dependency tree parsing for syntactic extractions. The detailed output is described below:

Relationship types - descriptions & examples

Our relation extraction combines dependency tree parsing and Semantic Role Labeling (SRL). We first design a set of patterns to mine relationship patterns from dependency trees. The patterns are extensions of two prior works: Ollie and ClauseIE. Second, we form extractions from SENNA’s Semantic Role Labeling (SRL) model. We combine dependency-based extraction techniques together with SRL to increase the recall of our system. Then we apply cleaning and deduplication techniques to select unique and high precision extractions.

The following table summarizes our relationship extraction patterns:

| Patterns                        | Patterns Type | Example                              | Derived Extraction          |
|---------------------------------|---------------|--------------------------------------|-----------------------------|
| (nsubj, verb, dobj)             | SVO           | Christie fired Kelly                 | (Christie, Fired, Kelly)    |
| (nsubj, verb (no obj), prep)    | SVP           | Wildstein resigned on Dec. 6th       | (Wildstein, resigned, on Dec 6th) |
| (nsubj, verb, noun-cop)         | SVCop         | The lane closures were retribution   | (The lane closures, were, retribution) |
| (nsubj, verb (with obj),        | SV(O)P        | Christie fired Kelly on Jan 8th      | (Christie, fired kelly on, Jan 8th) |
| prep) | The lanes were shut down for a traffic study | (The lanes, were shut down, for a traffic study) |
| (nsubj, verb)* | SV | Biden died of heart attack | (Biden, died) |
| (word, appos, word) | Appos | Christie fired that aide, Bridget Anne Kelly, a deputy chief of staff. | (Bridget Anne Kelly, is, a deputy chief of staff) |
| (A0, Verb, A1) (A0, Verb, A2) (A1, Verb, A2) | SRL | Ring was uncovered by the leaked Podesta emails dumped by Wikileaks | (by wikileaks, dumped, the leaked Podesta emails) (by the leaked Podesta emails dumped by wikileaks, uncovered, ring) |

*We take SV extraction only if subject does not come with an object or a complement, and the verb is among a set of predefined intransitive verbs such as die, or walk.

In addition to the above patterns, we have extended patterns such as (nsubjpass, verb, dobj), (xsubj, verb, dobj) by which we extract relationships from passive sentences. Also we extract relationships when there is a “conjunction and” is present. For example, from a sentence “Prosecutors have charged Kelly and Baroni.”, not only do we extract (Prosecutors, have charged, Kelly) as a SVO extraction, but also we retrieve (Prosecutors, have charged, Baroni) since the object (Kelly) is connected to another noun (Baroni) via a “conj_and” edge in the dependency tree.

Additional example sentences:
For Bridgegate:
1. Christie fired Kelly from her job in January after emails came to light connecting her to the closures.
2. Wildstein resigned on Dec. 6, calling the bridge scandal a distraction.
3. His administration initially claimed the lanes were shut down for a traffic study.
4. Bridget Kelly, the governor's former deputy chief of staff, told jurors in federal court in Newark that she discussed the plan to shut down access lanes at the George Washington Bridge with Christie.
5. The report places blame on Kelly, the deputy chief of staff Christie fired, and David Wildstein, whom Christie appointed to a post at the Port Authority.
6. Christie fired that aide, Bridget Anne Kelly, a deputy chief of staff.
7. Prosecutors have charged Kelly and former Port Authority of New York and New Jersey executive Bill Baroni, Christie's highest ranking political appointee at the transportation agency, with creating massive traffic gridlock in Fort Lee, New Jersey, as payback after the town's Democratic mayor, Mark Sokolich, refused to back Christie's 2013 reelection campaign.
For Pizzagate:
1. It claimed that Hillary Clinton and her campaign chief were running a child trafficking ring in the restaurants back rooms.
2. ring that was uncovered by the leaked Podesta Emails dumped by Wikileaks.
3. Biden died of heart attack, so did breitbart.
4. Democrats have alleged the lane closures were punishment directed at Sokolich for failing to endorse Christie in his reelection bid last year.
5. Democrats allege the lane closures were retribution against the mayor for failing to endorse Christie.

Example 1:
Sentence:
Christie fired Kelly on Jan. 8 after emails obtained by The Record showed she apparently ordered the lane closures.

Covers: SVO, SV(O)P, SRL
| Type    | Arg1          | Relation         | Arg2          |
|---------|---------------|------------------|---------------|
| SVO     | {Christie}    | {fired}          | {Kelly}       |
|         | {she}         | Apparently {ordered} | The lane {closures} |
| SV(O)P  | {Christie}    | {fired} <<{Kelly}>> on | {Jan.} 8     |
| SRL     | {Christie}    | {fired}          | {Kelly}       |
|         | By The {Record}| {obtained}      | {emails}      |
|         | {emails} obtained by The Record | {showed} | {she} apparently ordered the lane closures |
|         | {she}         | {ordered}        | The lane {closures} |

The entire set of results from running relation extraction can be found at:

Bridgegate - Link to the complete excel file: [https://drive.google.com/open?id=1ncrtvlflf5yZkEks3y0xf4wwPti9V9Yc](https://drive.google.com/open?id=1ncrtvlflf5yZkEks3y0xf4wwPti9V9Yc)
Pizzagate - Link to the complete excel file: [https://drive.google.com/open?id=1o8RU6dgwt08og7szO4F6X5T5-ORZmoFq](https://drive.google.com/open?id=1o8RU6dgwt08og7szO4F6X5T5-ORZmoFq)

**Block 10:** In this step we apply Named Entity Recognition (NER) tools to find various types of named entities along with their frequencies. In addition to NER, we used headword occurrences to find mentions of nouns as concepts in our corpus. The headwords are usually the words in arguments that happen to be at a higher level in the dependency tree. In SRL, we have the relations without a dependency tree. In order to tackle this problem, we find the corresponding word IDs in the parse tree and then pick the top node in the SRL argument to use as headwords for extractions. After combining the headwords and mentions, we have almost enough seeds to cover all the arguments in our relationship graph. Examples of NER types and the aggregated final list is shown below:

| NER Type | Definition                                      | Example                          |
|----------|------------------------------------------------|----------------------------------|
| 1        | PERSON                                         | People, including fictional      | Christie                        |
| 2        | ORG                                            | Companies, agencies, institutions | Port Authority                  |
| 3        | GPE                                            | Countries, cities or states      | Fort Lee                        |
| 4        | FAC                                            | Facilities such as buildings, airports, bridges | George Washington Bridge |
| 5        | NORP                                           | Nationalities or religious or political groups | Republican party |
In this step, we combine the headword and NER lists to get the most important and frequent mentions. It is interesting to note that the number of mentions of entities has a power law distribution for both Bridgegate and Pizzagate. More precisely, the log-log plot of frequency vs rank is a straight line, following Zipf’s law.
The top 100 aggregated mentions are printed below for each dataset:

Bridgegate - top 100

| rank | entity   | type      | frequency_score_sum_NER_arg |
|------|----------|-----------|-----------------------------|
| 1    | christie | PERSON    | 23925                       |
| 2    | port authority | ORG    | 7366                        |
| 3    | wildstein | PERSON   | 5785                        |
| 4    | fort lee  | GPE       | 4729                        |
| 5    | kelly     | PERSON    | 4284                        |
| 6    | baroni    | PERSON    | 3818                        |
| 7    | new jersey| GPE       | 3338                        |
| 8    | stepien   | PERSON    | 2718                        |
| 9    | sokolich  | PERSON    | 2496                        |
1. campaign  OTHER(ARG)  273
2. information  OTHER(ARG)  273
3. fifth amendment  LAW  272
4. dawn zimmer  PERSON  269
5. jersey  GPE  267
6. mowers  PERSON  266
7. director  OTHER(ARG)  263

Pizzagate - top 100:

| rank | entity       | type         | frequency_score_sum_NER_arg |
|------|--------------|--------------|-----------------------------|
| 1    | people       | OTHER(ARG)   | 2374                        |
| 2    | children     | OTHER(ARG)   | 1247                        |
| 3    | alefantis    | PERSON       | 890                         |
| 4    | clinton      | PERSON       | 885                         |
| 5    | reddit       | ORG          | 864                         |
| 6    | podesta      | PERSON       | 854                         |
| 7    | fbi          | ORG          | 843                         |
| 8    | trump        | PERSON       | 754                         |
| 9    | pizzagate    | PERSON       | 646                         |
| 10   | someone      | OTHER(ARG)   | 601                         |
| 11   | anyone       | OTHER(ARG)   | 596                         |
| 12   | evidence     | OTHER(ARG)   | 566                         |
| 13   | police       | ORG          | 551                         |
| 14   | vatican      | FAC          | 523                         |
| 15   | facebook     | ORG          | 484                         |
| 16   | anything     | OTHER(ARG)   | 454                         |
| 17   | haiti        | GPE          | 443                         |
| 18   | wikileaks    | ORG          | 441                         |
things
information
jews
hillary
something
washington
man
wikipedia
cia
catholic
voat
time
use
post
comet pizza
church
ring
alig
story
america
account
services
israel
way
american
subreddit
u
united states
|   |   |   |   |   |
|---|---|---|---|---|
| 47 | twitter | ORG | 316 |
| 48 | thing | OTHER(ARG) | 313 |
| 49 | guy | PERSON | 309 |
| 50 | msm | ORG | 308 |
| 51 | person | OTHER(ARG) | 308 |
| 52 | lot | OTHER(ARG) | 308 |
| 53 | hollywood | GPE | 301 |
| 54 | comet | ORG | 295 |
| 55 | pizza | ORG | 293 |
| 56 | james alefantis | PERSON | 291 |
| 57 | news | OTHER(ARG) | 286 |
| 58 | years | OTHER(ARG) | 282 |
| 59 | obama | PERSON | 282 |
| 60 | us | GPE | 281 |
| 61 | users | OTHER(ARG) | 273 |
| 62 | nothing | OTHER(ARG) | 272 |
| 63 | podestas | PERSON | 271 |
| 64 | investigation | OTHER(ARG) | 269 |
| 65 | case | OTHER(ARG) | 267 |
| 66 | comet ping pong | ORG | 260 |
| 67 | place | OTHER(ARG) | 258 |
| 68 | shit | OTHER(ARG) | 257 |
| 69 | kids | OTHER(ARG) | 253 |
| 70 | finders | ORG | 252 |
| 71 | abuse | OTHER(ARG) | 248 |
| 72 | site | OTHER(ARG) | 245 |
| 73 | trademarks | OTHER(ARG) | 242 |
| 74 | fbianon | ORG | 239 |
Blocks 12, 13: A large fraction of mentions of entities co-occur with each other. For example, ‘hillary’ and ‘clinton’ repeat in a high number of arguments. As explained in the main paper, grouping the most co-occurrence mentions in the same context helps to form initial nodes with
higher resolution. For example, if we group ‘hillary’ and ‘clinton’ in the same contextual group, then, ‘Clinton foundation’ and ‘hillary foundation’ will be in the same subnodes. We call this group of words in the same context a supernode. The goal of this process is to create supernodes which carry the most relevant arguments. For example ‘pizza owner’ and ‘comet owner’ are assumed to be in the same subnode at a certain granularity, since comet pizza is a frequent term in the pizzagate corpus. Therefore, if we contextually group these terms in the same supernode, we place them in the same subnode. For this purpose, we used a maximum number for limiting the number of mentions in each supernode. Here, we used 4 as the maximum number of entities to be contextually grouped to form the seed nodes in any supernode. We removed stopwords after the formation of supernodes. The results of this step are shown in step 13.

The supernode seed list for bridgegate:

| nodeID | nodeLabel |
|--------|-----------|
| 0      | authority port executive |
| 1      |christie governor chris former |
| 2      |wildstein david |
| 3      |kelly bridget anne |
| 4      |new jersey york |
| 5      |lee fort mayor sokolich |
| 6      |baroni bill |
| 7      |democrat democrats democratic |
| 8      |stepien |
| 9      |bridges bridge george washington lane |
| 10     |samson |
| 11     |recordings records |
| 12     |bills billing |
| 13     |office officer |
| 14     |governors |
| 15     |mastro randy attorney |
| 16     |wisniewski john assemblyman dean |
| 17     |attorneys |
| 18     |michael michaels critchley |
| 19     |gwb |
| 20     |state states house weinberg loretta |
| 21     |zimmer |
| 22     |ann vardeman pollster monmouth |
| 23     |closures closure |
| 24     |committees legislative panel state |
| 25     |reporters reporting reports mastro |
| 26     |foye |
| 27     |dunn gibson crutcher lawyer |
allege allegations alleged allegation testimony testimonial baroni client clients firm managers management manager manage campaign rutgers company companies private republican republicans hoboken mayors rena christina genovese gop legislatures legislature baldassare sweeney stephen marino kevin affairs affair intergovernmental fulop academy jose marti freshman msnbc presidency bush jeb candid candidate candidates presidential boburg shawn wigenton susan unions union rockefeller judges verizon lesniak raymond cortes garten

Supernode Seed lists for for pizzagate:

| nodeID | nodeLabel                  |
|--------|----------------------------|
| 0      | podesta                    |
| 1      | pizza comet ping pong      |
| 2      | parties party              |
| 3      | clinton hillary            |
| 4      | ring pedo                  |
| 5      | child                      |
| 6      | pizzagate                  |
| 7      | trump                      |
| 8      | barack obama               |
Block 14: In this step, we used the method described in the main paper to ‘measure’ the importance of a verb occurring between two target actants. For each pair of two supernodes, we used the dependency tree to find the verbs which occur in sentences that include any mention of each supernode seed word. As explained in more detail in the paper, we assign a significance score to each verb given two supernodes. Therefore, the output is a memory based dictionary where, with the key of two target supernodes, we get a score for each verb.

Block 15: As part of the process of generating the automated graph, we collect all arguments which have at least one mention corresponding to each of the supernodes. This provides an initial form of the graph, where supernodes constitute the nodes and the corresponding relations constitute the edges. This interim graph is very noisy. For example, ‘hillary clinton’ and ‘hillary campaign’ are mapped to the same node. In order to solve this problem, we use embeddings that encode the semantics of phrases. With the help of an unsupervised k-means algorithm, we broke down the supernodes into smaller sized and more dense nodes, which we call subnodes. With this approach, we can also tune the number of clusters.

Block 16: Given the output of the k-means clustering, we have the set of subnodes.

Block 17: In order to find the best words describing each subnode, we use a scoring similar to TFIDF, where the documents are the paragraphs and the term frequencies are based on the subnode arguments. Given this score, we choose up to five top scored words.

Block 18: After finding the best word or words to describe each subnode, we merge the subnodes based on similar labels. In order to reduce the number of subnodes, we apply a thresholding based on the relative size of subnodes to the average size of nodes for every supernode. Shown below are plots for an example supernode from each corpus, pizzagate and bridgegate. In the log-log plot, the threshold is sharp:
Subnode list (as defined by their automated labels) for bridgegate:
nodeID nodeLabel
0 authority port
1 Chris christie
2 Samson
3 chairman samson
4 former
5 David Wildstein
6 christie painting
7 Bill Stepien
8 Joe Peyronnin
9 Bridget Anne Kelly
10 New Jersey
11 New York
12 Mark Sokolich
13 Fort Lee Mayor
14 Bill Baroni
15 Lane Closure gwb
16 gwb Bridge Lane
17 Lane
18 Wolff Samson
19 file record
Stephen Sweeney
state weinberg
chris peyronnin
hoboken
lawyer
Clinton
president
Regina Egea

Subnode list for pizzagate:
nodeID nodeLabel
0 tony podesta
1 pizzagate pizza
2 party
3 article part party art
4 martin stewart party
5 podesta
6 podesta podestas
7 pizzagate
8 alefantis
9 clinton
10 pedophile pedophilia
11 pedo pedos
12 pedo ring
13 child
14 trafficker trafficking child blackmail dead human
15 trafficking
16 tunnel
17 wikileaks email wiki wikipedia podesta manager
18 wikipedia wikileaks pedophile
19 email
20 email podesta leaked
21 twitter
22 satan
23 part party art article artist partner
24 instagram
25 comet story pizza
26 comet
28 pizza pizzagate
29 pedophile
30 obama
31 email single
32 article artist party participant art partner
33 handkerchief
34 satanic
35 part heart art
36 clinton contribution foundation
37 ring hearing
38 wikileaks
39 satanism satanist satanic
40 jimmycomet comet james alefantis
41 pizza
42 wikipedia
43 james alefantis
44 hillary clinton
45 part
46 party labour
47 child porn
48 clinton foundation
49 pizza comet
50 john podesta
51 hillary
52 instagram photo
53 facebook
54 pong ping comet
55 trump
56 podesta john clinton campaign hillary chairman
59 james alefantis ,instagram ,owner ,account ,comet ,
60 satanic ,ritual ,
61 cannibal ,openly ,cp ,posting ,sick ,pedo ,
62 alefantiss ,
63 trafficking ,human ,sex ,slave ,
64 pedophilia ,pedophile ,
65 satanism ,
66 tunnel ,underground ,sewage ,
67 donald ,trump ,
68 party ,casa ,podesta ,chicago ,mahababani ,pizza ,
69 podestas ,
70 obamas ,advance ,team ,personal ,president ,
71 wl ,
72 rothschild ,child ,childrenkanye ,childporn ,16 ,12 ,
73 hovered ,satan ,smoke ,vatican ,
74 party ,labour ,child ,birthday ,councillor ,photo ,
75 trump ,epstein ,hosted ,attended ,party ,employee ,
76 ring ,
77 hunting ,trip ,scalia ,
The Choice of Thresholds in determining Core and Overlapping Nodes in Community Partitioning of Narrative Networks:

As described in the main paper, (i) We first perform community detection $T_{max}$ times for a given narrative network, (ii) Then, we create a community co-occurrence matrix $A(i,j) = k$, if nodes $i$ and $j$ co-occur $k$ times in $T_{max}$ runs, (iii) Finally, we normalize it by $T_{max}$ to get the probability that a pair of nodes $(i,j)$ co-occur in any random run. If each run returned the exact same community structure, then the normalized matrix will have only 0 or 1 values, forming an adjacency matrix. The resulting graph defined by such an adjacency matrix will be a set of cliques, where each clique represents the nodes belonging to the same community. However, due to randomness in the community partitioning algorithm, loosely connected nodes often change community assignments, and hence the normalized co-occurrence matrix has values $0 \leq A(i,j) \leq 1$. Next we use a threshold value $P_{th}$ to create an adjacency matrix as follows: If $A(i,j) < P_{th}$ then we set it 0, otherwise if $A(i,j) \geq P_{th}$ then it is set to 1. Next we find the connected components in this graph and record the size of the Giant Connected Component (GCC) and the number of connected components. The thresholding procedure could create disconnected and isolated nodes, and we disregard such isolated nodes, as described in Algorithm 1 in the main section.

The following plot shows the case for the Pizzagate narrative network ($T_{max} = 1000$), recording both the GCC size (the left Y-axis), and the number of connected components (the right Y-axis) as a function of $P_{th}$. As one can observe, when $P_{th}$ is set to 1, the GCC size is minimum, and as it is decreased to 0 it becomes the entire network. In between, it increases in steps, as connected components get merged. Similarly, as the threshold is decreased the number of connected components decreases until it becomes a single connected component. As one can
see from the plot, the connected components remain quite separated until around $P_{th} = 0.45$ when many of them merge to create a large connected component.

To determine a good threshold to get the core nodes, i.e. determining $P_{th_1}$, we need to meet two conditions: (i) The GCC size should not increase as the threshold is decreased from $P_{th_1}$; in other words the GCC size has a long flat region to the left of $P_{th_1}$, and (ii) The number of connected components does not change as well, and the plot should have a corresponding flat region. As shown in the above figure this is satisfied if we set $P_{th_1} = 0.7$. The oval highlights point out the corresponding flat regions. Each connected component defines a set of core nodes. Figs. 10 and 8 in the main paper correspond to core nodes determined by $P_{th_1} = 0.7$.

In order to determine overlapping nodes, we consider a value of $P_{th}$ where the GCC size has increased sharply, indicating that a lot of the sparsely connected nodes are being absorbed by the core communities. We pick $P_{th_2} = 0.4$ where the GCC size has a flat region.

Fig. 10 in the main paper corresponds to these choices of $P_{th_1}$ and $P_{th_2}$. 
For bridgegate, we again picked $T_{max} = 1000$ and the corresponding plots of GCC size and the number of connected components are shown below. As one can see, the GCC size has a sharp transition over a small region: $0.65 < p_{th} < 0.72$ where it goes from being a small structure to absorbing pretty much all the nodes. This is an indication of a so-called phase transition, where a small shift in threshold leads to a huge change in connectivity.

If one picks the same thresholds as in Pizzagate, then as shown in the following figure, the resulting community structure is dominated by one large community and everything gets connected via the overlapping nodes.
