

S1 Appendix

A. The procedure of creating training data

The procedure used in the section ‘Data collection and training data for opinion identification’ is described as follows. It includes five steps:

1) Identifying seed hashtags. Among the most frequent hashtags, we identify four hashtags, each one representing a different opinion category: #maga for pro-Trump (the abbreviation of the official Trump campaign slogan: Make America Great Again), #imwithher for pro-Hillary (the official Clinton campaign slogan), #nevertrump for anti-Trump, and #neverhillary for anti-Clinton.

2) Building co-occurrence networks of hashtags. The hashtag co-occurrence network $H(V, E)$ is then constructed, where the set of vertices $v_i \in V$ represents hashtags, and an edge e_{ij} is drawn between v_i and v_j if they appear together in a tweet. The resulting graph has 108,600 vertices and 881,344 edges.

3) Computing similarity networks of hashtags. With the constructed $H(V, E)$, we test the significance of each edge e_{ij} [1]. Specifically, the probability p_{ij} (p-value of the null hypothesis) to observe the corresponding number of co-occurrences k by chance is computed given the number of occurrences n_i and n_j of the vertices v_i and v_j , and the total number of tweets N (Equation 1). Only significant edges satisfying $p_{ij} < p_o$, where $p_o = 10^{-6}$, are kept, effectively filtering out spurious relations between hashtags. Then, a weight s_{ij} is assigned to the edge e_{ij} (based on Equation 2) representing the significance of the similarity

between two hashtags. Retaining only significant similarity linkages reduces the graph to 108,600 vertices and 51,633 edges, which comprises a similarity network of hashtags.

$$p_{ij}(k) = \prod_{m=0}^{n_j-k-1} \left(1 - \frac{n_i}{N-m}\right) \prod_{m=0}^{k-1} \frac{(n_i-m)(n_j-m)}{(N-n_j+k-m)(k-m)} \quad (1)$$

$$s_{ij} = \log \left(\frac{p_o}{p_{ij}} \right) \quad (2)$$

4) Classifying hashtags in the similarity network. The similarity network is then used to discover and classify hashtags that are significantly similar to the seed hashtags as in [2,3].

The main idea behind the classification of hashtags is that a vertex v_i determines its label, denoting the class to which it belongs, based on the labels of its neighbors, defined as vertices that have edges with v_i . It is assumed that each v_i in the network chooses to obtain the label carried by the largest number of its neighbors, with ties broken uniformly randomly.

Formally, if C_1, \dots, C_j are the labels that are currently active in the network and $d_i^{C_j}$ is the number of neighbors v_i has with label C_j , then every v_i keeps acquiring its label until for each v_i ,

$$\text{If } v_i \text{ has label } C_m, \text{ then } d_i^{C_m} \geq d_i^{C_j} \quad (3)$$

5) Network pruning and human validation. We further prune the resulting set of labelled hashtags by retaining only those that meet the following criterion.

$$n_i > r \max_{v_j \in C_m} n_j \quad (4)$$

where n_i is the occurrences of the hashtag associated with vertex v_i , C_m is the label of v_i and $r = 0.001$ is a threshold parameter. Finally, discovered hashtags are human validated to select hashtags having direct reference to the candidate, its party or slogans of the candidate and that express an opinion. In practice, the entire process (steps 1-5) is iterative until a stable

set of hashtags is found. Table 1 shows the top-10 labelled hashtags with the highest occurrence for each category. The final network results in four different clusters corresponding to the Pro-Clinton, Anti-Clinton, Pro-Trump and Anti-Trump hashtags (Fig 1).

Table 1 Top-10 labelled hashtags with the highest occurrence for each category

Pro-Clinton	Anti-Clinton	Pro-Trump	Anti-Trump
#imwithher	#dickileaks	#trump Pence16	#dumpthetrump
#clintonkaine	#hillaryforprison2016	#trump rally	#nevertrump
#hillary2016	#neverhillary	#trumpstrong	#lockhimup
#votehillary	#crookedhillary	#draintheswamp	#bullytrump
#imwithher2016	#hillarysemails	#maga3x	#nastywomenunite
#hillaryforamerica	#lockherup	#maga	#racist
#hillaryclintonforpresident	#corrupthillary	#trump2016	#stoptrump
#hillarysarmy	#clintonscandals	#votetrump	#loserdonald
#momsdemandhillary	#indicthillary	#makeamericagreatagain	#lyingtrump
#womensvote	#notwithher	#americafirst	#anybodybuttrump

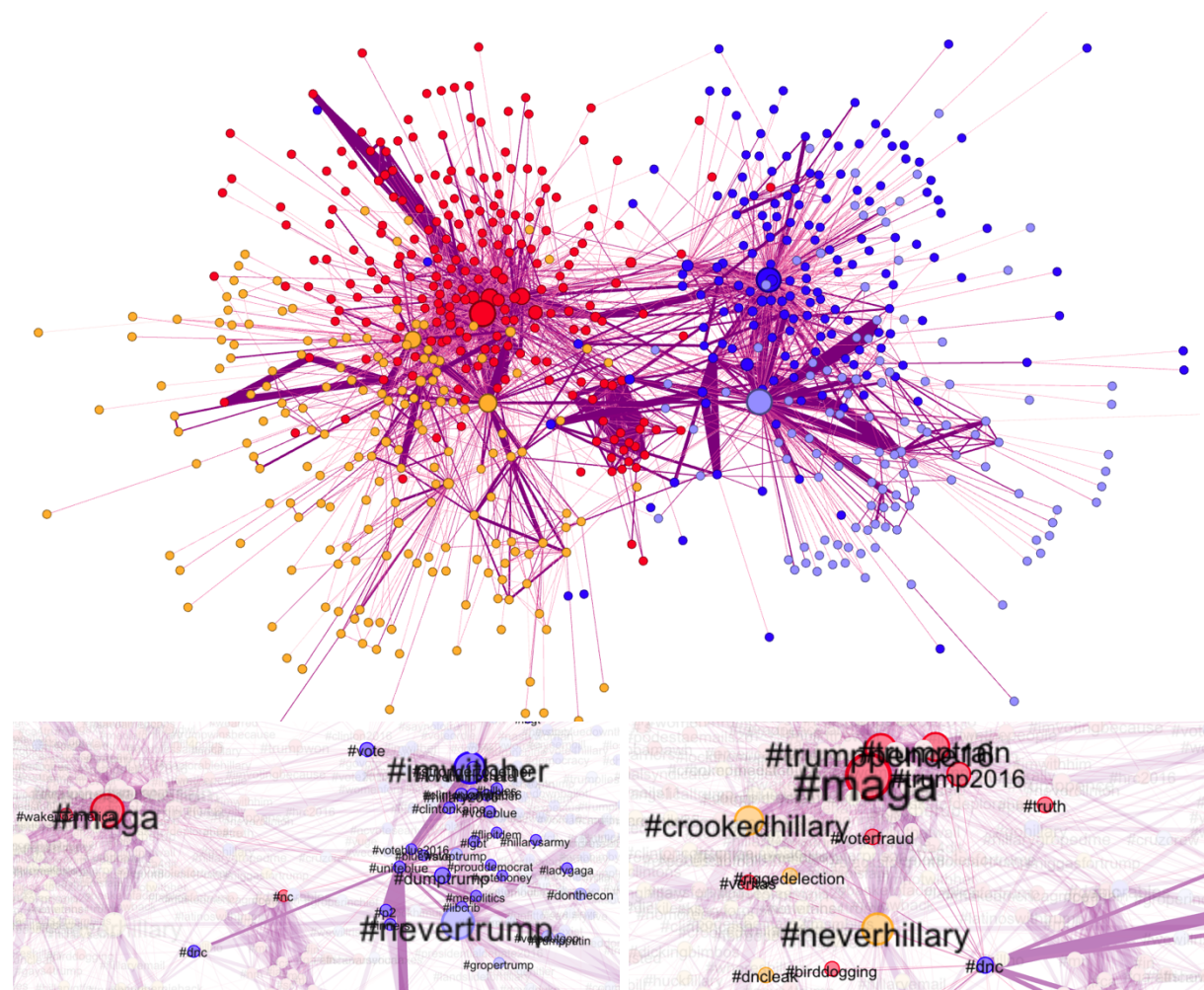


Fig 1. Clusters of hashtags. Red: Pro-Trump, Orange: Anti-Clinton, Dark Blue: Pro-Clinton, Light Blue: Anti-Trump.

Once a set of labelled hashtags is generated, we use them to classify tweets into opinion categories by counting number of hashtags for each label and assign the most common label to the tweet. The following 8 opinion categories are generated: Pro-Clinton, Anti-Trump, Support-Clinton (with Pro-Clinton and Anti-Trump two most common labels), Pro-Trump, Anti-Clinton, Support-Trump (with Pro-Trump and Anti-Clinton two most common labels), mixed (with one of Pro-Clinton and Anti-Trump and one of Pro-Trump and Anti-Clinton as two most common labels), and unidentified (no labelled hashtag). We use only tweets for the first 6 categories, in total 238,142, for training.

B. Relative opinion measure at the word and user levels

As the opinion-oriented word embedding measures relative opinion at the word level, we examine the trained embeddings for the labelled hashtags to evaluate model performance at the word level. Fig. 2 plots a two-dimensional Multi-Dimensional Scaling (MDS) representation of the hashtags in Table 1. The four seed hashtags (#maga for pro-Trump, #imwithher for pro-Hillary, #nevertrump for anti-Trump, and #neverhillary for anti-Clinton) highlighted by black circles showing a bipolarized pattern. Fig. 2 shows that the two groups, Pro-Trump/Anti-Clinton and the Pro-Clinton/Anti-Trump, of hashtags are clearly separated from one another (the dash line shows a rough dividing boundary for the two groups), while within each group the two categories of hashtags are mixed together. It indicates that our trained embeddings correctly perform with learned opinion orientations.

Furthermore, the relative opinion measure at the user level can be obtained by aggregating word embeddings first to the tweet document level then to the user level. To evaluate the performance of user-level relative opinion measure, we selected the user accounts for the two candidates, Donald Trump and Hillary Clinton, whose opinion orientations are publicly known. Fig. 3 shows a two-dimensional MDS representation of the aggregated opinion embeddings for the two candidates and the word embeddings of the four seed hashtags as references. The four seed hashtags are bipolarized between Pro-Trump/Anti-Clinton and the Pro-Clinton/Anti-Trump, as they are trained embeddings (in-sample prediction). The two candidates' user accounts can be reasonably separated and they are toward the expected opinion orientations respectively according to their distances to the four opinion extremes identified by the seed hashtags. As the two candidates' opinion embeddings are inferred (out-of-sample prediction), their separation is not as extreme as the bipolarization of seed hashtags. Rather, they show some similarity, because both user accounts have posted tweets that include hashtags belonging to the opposing opinion categories for the purpose of attack or criticizing.

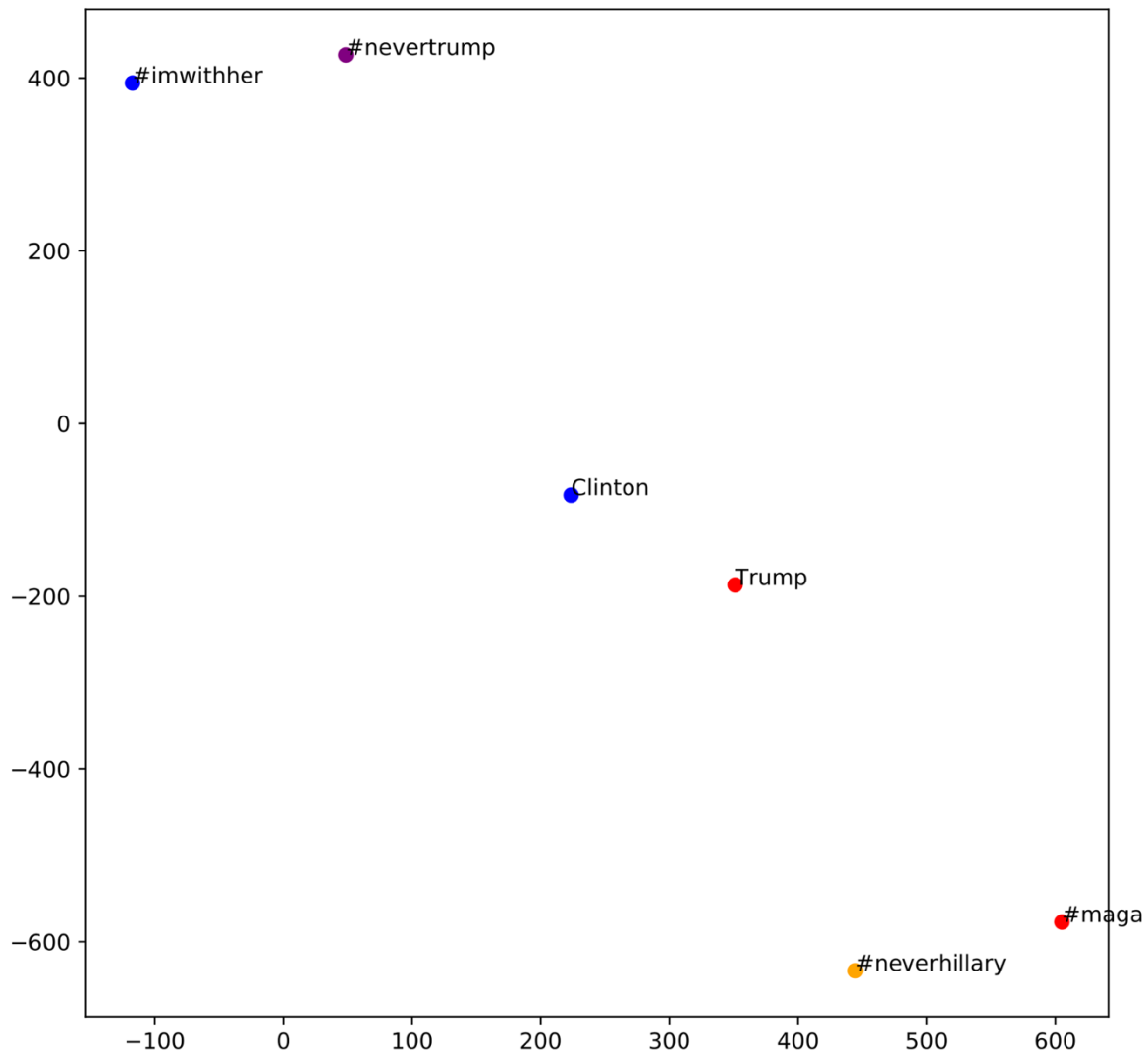


Fig 3. Clusters of hashtags. Red: Pro-Trump, Orange: Anti-Clinton, Blue: Pro-Clinton, Purple: Anti-Trump.

References

1. Martinez-Romo J, Araujo L, Borge-Holthoefer J, Arenas A, Capitán JA, Cuesta JA. Disentangling categorical relationships through a graph of co-occurrences. *Phys Rev E*. 2011;84: 046108. doi:10.1103/PhysRevE.84.046108
2. Bovet A, Morone F, Makse HA. Validation of Twitter opinion trends with national polling aggregates: Hillary Clinton vs Donald Trump. *Sci Rep*. 2018;8: 8673. doi:10.1038/s41598-018-26951-y
3. Raghavan UN, Albert R, Kumara S. Near linear time algorithm to detect community structures in large-scale networks. *Phys Rev E*. 2007;76: 036106. doi:10.1103/PhysRevE.76.036106

