Topical Analysis of Twitter User Clusters

Mara Schwartz (schwam4@rpi.edu), Tomek Strzalkowski (tomek@rpi.edu)

Dataset

2017 French Election twitter dataset.

#Tweets			#Users
674309			59144
#Retweets	#Replies	#Tweets	
504526	116676	53107	

Retweets are used to create digraph of users. User *u* has an edge to user *v* if *u* retweeted *v*. The weight of the edge, *e* = (*u*,*v*), is the number of times *u* retweeted *v*. Some users were located outside of the dataset so there is minimal information on the tweet.

#Nodes		#Edges	
65517	Overlap With Dataset	222231	Weighted
	46683		504526

Method

- Tweets are translated using the Helsinki-NLP opus-mt-fr-en model and Hugging Face transformers
- The subset of retweets are used to create a digraph of twitter users.
- The graph is clustered using the Leiden Clustering algorithm after comparing several different clustering algorithms to determine the best one for this medium.
- We turn each cluster into a conversation and perform topical analysis on the conversation, extracting meso-topics (the most prevalent topics in a dialogue).
- Hashtags are also analyzed to give more insight into what each cluster is about.

Clustering Algorithms

Louvain. This algorithm is a modularity optimization algorithm comprised of two steps. (1) Local moving of nodes and (2) Aggregation of nodes.
Leiden. This expansion on the Louvain algorithm is comprised of three steps. (1) Smart local moving, (2) Refinement of the partition, and (3) Aggregation of nodes. Smart local moving is a more efficient version of local moving where only nodes whose neighborhood has changed are visited. Partitions created are refined often leading to clusters being split into subclusters.
Label Propagation. Every node is initialized with a unique label. At every iteration, each node updates its label to the most common one amongst its neighbours.
Info Map. Uses probability flow of random walks on a network as proxy for information flows. Minimizes the length of the description of the probability flow, defined as the entropy of movement between modules + entropy of movement within modules.
Markov Clustering. Computes probability of random walks through the graph using two operators, expansion and inflation.

Comparing Clustering Algorithms

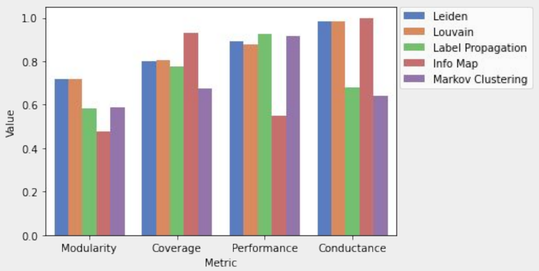
Here is a summary of each algorithm, the time it took to compute the partition, the number of clusters, and the node and edge coverage of the largest 10 clusters.

Algorithms	Compute Time (s)	#Clusters	10 Largest Clusters	
			%nodes	%edges
Leiden	0.19	1622	89.74%	96.31%
Louvain	5.55	1630	90.93%	97.12%
Label Prop	2.69	4648	67.10%	81.93%
Infomap	2.3	1479	94.83%	99.06%
Markov	17	4615	68.34%	70.21%

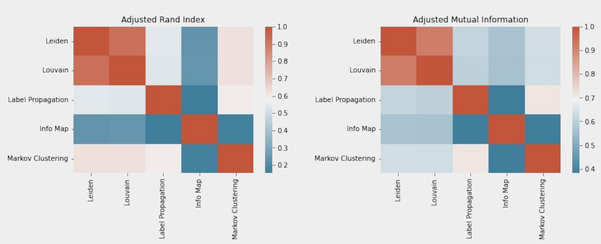
Infomap had the highest coverage, but this doesn't necessarily mean it was better. Infomap had only 3 clusters with more than 1000 users. Leiden and Louvain had 9 and 8 respectively and better represented the data.

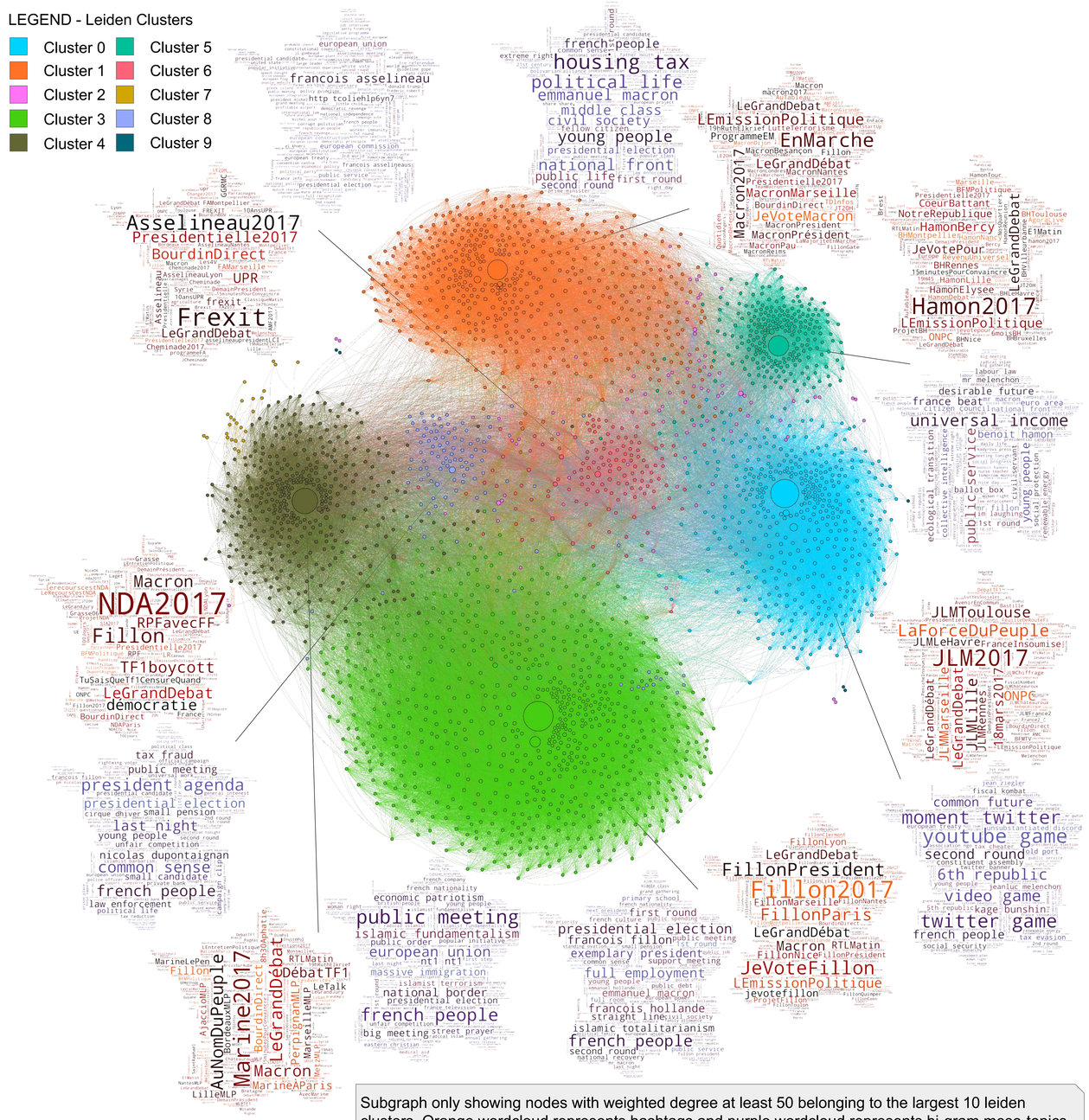
Further evaluate the algorithms on the following metrics

- **Modularity.** The ratio of intra-cluster edges of a partition with the expected number of such edges given by a Newman-Girvan null model.
- **Coverage.** Ratio of intra-cluster edges to total number of edges in the graph.
- **Performance.** Defined as (intra-cluster edges + inter-cluster non-edges) / total potential edges.
- **Conductance.** For a given cluster S, conductance is defined as the CutSize(S) / min(|S|,|G-S|). The conductance for a full partition is the mean of the conductance for each cluster.



- **Adjusted Rand Index (ARI).** The RI computes a similarity measure between two partitions by counting all pairs of nodes assigned in the same or different clusters. It's adjusted for chance with the expected RI.
- **Adjusted Mutual Information (AMI).** Measures how much we know about partition B given partition A. Partitions with a high level of MI are more similar. This metric is adjusted for chance similarly to the ARI.





Subgraph only showing nodes with weighted degree at least 50 belonging to the largest 10 leiden clusters. Orange wordcloud represents hashtags and purple wordcloud represents bi-gram meso-topics

Future Work

- More detailed socio-linguistic analysis of the generated conversations including Topic Control.
- Auto-tagging of dialogue to allow for further socio-linguistic analysis to be performed.
- Statistical inter-cluster comparison showing that the clusters are sufficiently distinct in semantic content. Could include sentence similarity measurements to show that users show more similarity with the clusters they belong to than to clusters they don't belong to.

References

Emmons, Scott & Kobourov, Stephen & Gallant, Mike & Borner, Katy. (2016). Analysis of Network Clustering Algorithms and Cluster Quality Metrics at Scale. PLOS ONE. 11. 10.1371/journal.pone.0159161.

Fortunato, Santo. (2009). Community Detection in Graphs. Physics Reports. 486. 10.1016/j.physrep.2009.11.002.

Traag, V. & Waltman, L. & van Eck, Nees Jan. (2019). From Louvain to Leiden: guaranteeing well-connected communities. Scientific Reports. 9. 5233. 10.1038/s41598-019-41695-z.

Waltman, Ludo & van Eck, Nees Jan. (2013). A smart local moving algorithm for large-scale modularity-based community detection. European Physical Journal B. 86. 10.1140/epjb/e2013-40829-0.