**BACKGROUND & PROBLEM STMT**

- Echo chambers: communities that share content with similar viewpoint; rarely interact with other communities in positive terms.
- Studies that have focused on echo chambers.
  - Use real network of follower/friends, but:
  - Complete network info is hard to obtain.
  - Labels may be biased themselves.
  - Labels may change over time, e.g. many former UKIP supporters voted for Labour in UK's general election of 17.
- After echo-chambers are identified, all of their content is dismissed as biased.

**Problem**
- Identify echo-chambers on Twitter, such that:
  - Network info is not required.
  - No human labeling is required.
  - Model can identify **which topics are subject to misrepresentation** by each echo-chamber.
- Do all of this in a way that the model can be updated in real-time.

**DATA & METHODOLOGY**

- Download all tweets that mentioned those domains within the same time period: 25,742,785 tweets from 131,552 unique accounts.
- Use PMI-like measure to score domains by:
  - similarity to each other.
  - similarity to Kaggle domains.
- Score users based on the domains they share:
  - Bias: similarity of a given user’s domains to Kaggle domains.
  - Partisanship: similarity of a given user’s domains to each other.
- Draw network of co-shared domains:
  - Node size: partisanship of each user.
  - Edge width: number and cumulative bias of two nodes’ shared domains.
- Run Louvain modularity detection to identify echo-chambers (color-coded).
- Calculate each chamber’s deviation:
  - Word2Vec deviation from a large global set.
  - TF-IDF deviation from other chambers.
- High Word2Vec deviation AND low TF-IDF deviation ⇔ potentially misrepresented topic.

---

**EVALUATION**

- **Buzzfeed’s list of 60 true stories and 60 fake stories** propagated on Facebook in the period leading up to 2016 U.S. presidential election.
- **No seed set ⇔ no bias score.**
- Apply echo-chamber analysis to the users.
- Train linear classifier based on: user partisanship, domain exclusivity, content match against each chamber’s set of deviating terms.
- F1 scores show the features are robust & informative, even for simple classifiers.

<table>
<thead>
<tr>
<th>Features</th>
<th>LogReg</th>
<th>DTree</th>
<th>SVMLinear</th>
</tr>
</thead>
<tbody>
<tr>
<td>user + domain</td>
<td>0.73</td>
<td>0.71</td>
<td>0.74</td>
</tr>
<tr>
<td>user + content</td>
<td>0.80</td>
<td>0.87</td>
<td>0.89</td>
</tr>
<tr>
<td>domain + content</td>
<td>0.75</td>
<td>0.70</td>
<td>0.75</td>
</tr>
<tr>
<td>All</td>
<td>0.77</td>
<td>0.73</td>
<td>0.78</td>
</tr>
</tbody>
</table>

---

**DISCUSSION**

- **Suggested setup for real-time setting:**
  - Network of chambers reconstructed on a daily/weekly basis.
  - An index constructed to score domains based on their popularity in each chamber.
  - Tweets stream from the public API (e.g. using Spark streaming).
  - Stories are tracked by their URL.
  - Users are benchmarked against the network.
  - To avoid exhausting rate limits, a KB of users’ common domains can be stored in a cache.
  - Each story’s score based on:
    - User scores
    - Content score
    - Domain score

---

**SEE WHERE YOU STAND**

**TRY THE LIVE DEMO**