2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining

Identifying Policy Agenda Sub-Topics in Political Tweets based on Community Detection

Rohit Iyer
Dept. of Computer Science
Iowa State University
Ames, Iowa 50011
Email: riyer@iastate.edu

Johnny Wong
Dept. of Computer Science
Iowa State University
Ames, Iowa 50011
Email: wong@iastate.edu

Wallapak Tavanapong
Dept. of Computer Science
Iowa State University
Ames, Iowa 50011
Email: tavanapo@iastate.edu

David A. M. Peterson
Dept. of Political Science
Iowa State University
Ames, Iowa 50011
Email: daveamp@iastate.edu

***Abstract*—The explosive use of twitter in the political landscape presents new avenues for tracking political conversations at federal and state level. Tweets are used by state and federal government bodies to present citizens with information about future and present policies. It is also used by political candidates to express their views on policy changes, laws and to campaign for legislative body elections, the most recent example being the 2016 US presidential elections. In this paper, we use supervised learning, textual semantic similarity and community detection techniques to find actively discussed policy agenda sub-topics among political tweets within a certain time period. Specifically, we target tweets pertaining to major policy agendas published by state representatives in US, to try and discern the major policy sub-topics that they address using their twitter accounts. Using our method, we demonstrate how we achieve a high accuracy in terms of Topic Recall and Order Recall, by comparing the output of our proposed method with sub-topic annotations done by domain experts.**

***Keywords*—twitter, Policy Agenda, Sub-topics, community detection, Convolutional Neural Networks, Semantic Similarity**

I. Introduction

The impact of twitter on the political landscape is a pressing topic of study. With increasing use of twitter by federal and state representatives to publish their political stand, people and news organizations are increasingly following major policy related discussion using the twitter feed of such politicians, including the US president Donald Trump. Twitter is being used to initiate an active discussions on immediate problems and concerns about federal and state laws and policies. Representatives in US are using twitter to express their views on critical policies like healthcare, immigration, defense etc. Consequently, it becomes an interesting problem to ascertain the major policies being addressed on twitter by such representatives during a given time period. Solving this problem helps ongoing research in political science, where researchers require a clear summary of the topics being addressed in important governmental institutions for problems. It also serves as a base for sub-topic recognition in other fields like finance, where entities rely heavily on sub-topic detection in huge document clusters.

In our work, we use tweets from the twitter feed of US state representatives as our source for extracting keywords depicting major policy agenda sub-topics being discussed. To achieve this, we use Convolutional Neural Networks (CNN), text-based similarity metric and community detection (shown in Figure 1).We now present a formal problem statement:

A. *Problem Definition*

We tackle the problem of identifying actively discussed sub-topics for each class of major policy agenda in a given time frame, from among tweets published by credible sources. For our experiment, we focus on tweets published only by twitter handles of state representatives in US. This helps us restricts our experiment to tweets which are uniform (in terms of their format), politically oriented and relatively clean in language. This ensures that the trained CNN classifier gives better results as the training set provided is uniform. Our work does not require any seed terms to recognize sub-topics. The user only needs to specify the time period and the name of the state that they would like to focus on.

Essentially, our process is to first form clusters of tweets sourced from the twitter feed of state representative, where each cluster addresses a policy agenda sub-topic. Once all tweets are processed, the next step is to iterate over each cluster and use tweets from within the cluster to extract significant keywords which define the cluster. This entire process can be represented as the following series of steps:

- Classify tweets among multiple high-level topic-based categories (where each category defines broad legislative topic like immigration, education etc.)
- For each such legislative topic, construct a tweet-similarity graph based on similarity metrics defined later
- Applying a weighted community detection methodology to stitch together groups of highly similar tweets.
- Find out the trending hash-tags and topical keywords for each sub-topic (represented by a community).

In short, the main contributions of our work can be summarized as follows:

- We present a way to use CNN and similarity model to represent political tweets spanning a time period to be represented in the form of a network graph.
- We show how we can use community detection on such a graph to find clusters of highly similar tweets based on the topic they address.

- We show how we can use results from community detection to find major sub-topics being discussed with high precision, in terms of topic recall.

Section II presents some of the past work related to extracting important keyword from text based content. Section III explains in detail about our proposed work on extracting keywords representing policy agenda sub-topics. Section IV summarizes the results obtained, based on our test set and compares it with policy sub-topic labelling achieved by domain-aware participants. Lastly, section V concludes our work with possible applications and possible future enhancements.

II. Related Works

There has been some work done on detecting trending topics in a collection of textual documents.

Document-centric methods exploit some type of similarity metric between documents. The work done by Phuvipadawat and Murata [1] approaches the problem of detecting breaking new topics in twitter using the above approach. Tweets which are retrieved using specific queries and hash-tags are converted into a bag-of-words form. Tweets are then assigned to clusters based on textual similarity between incoming tweets and existing clusters. Dimensions other than text have been used to give better cluster quality. [2] uses both text and temporal distribution to output trending topics. Such approaches suffer from noise sensitivity and fragmentation of clusters. To reduce these problems, manual selection of information providers are needed.

Feature-centric methods are based on statistical models to extract set of terms that represent a topic in a given set of documents. Most approaches are based on LDA (Latent Dirichlet Allocation) [3] and some extensions of LDA [4]. These approaches identify a set of bursty items and then use these items to define clusters defining topics.

Graph based approaches detect important keywords based on their pair-wise similarity score. The work done by Sayyadi et al. [6] creates a term co-occurrence graph, where each node represents a token and an edge depicts occurrence of 2 words/tokens in the same tweet and uses community detection to create topical clusters. However, this method focuses on individual tokens instead of entire sentences, losing out on some of the contextual information.

Apart from topic detection, there have also been some work done on inferring political opinion using twitter. [8] use natural language processing and sentiment analysis to detect political orientation in terms of sentiment[positive/negative/neutral] and policies[liberal/conservative/neutral]. Though some of the research works mentioned above have tried to recall ’trending’ topics based on short bursts of tweets, none of them have focused on policy agendas. Specifically, as the tweet corpus increases, their results become convoluted with inter-mixing of keywords across topics. With a huge corpus of tweets talking about policies, extracting meaningful discussion on policies has not been well handled, mainly due the possibly huge data-sets.

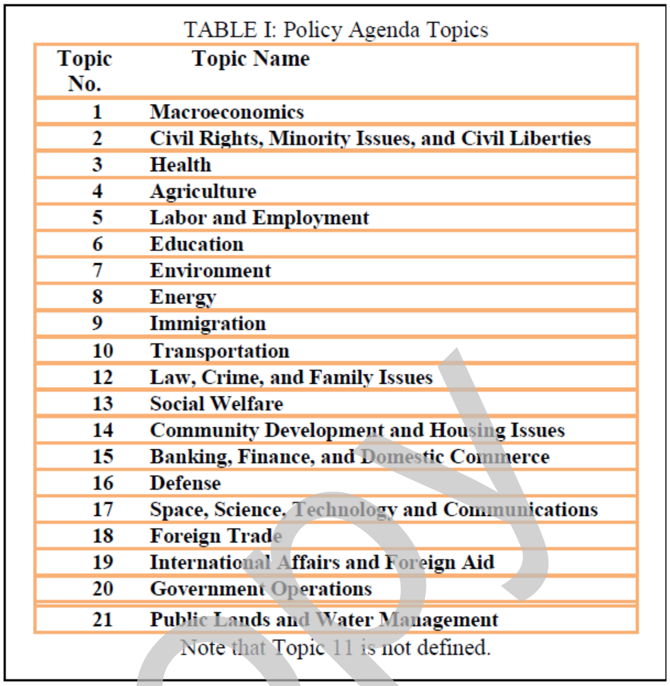

TABLE I: Policy Agenda Topics

Topic No.	Topic Name
1	Macroeconomics
2	Civil Rights, Minority Issues, and Civil Liberties
3	Health
4	Agriculture
5	Labor and Employment
6	Education
7	Environment
8	Energy
9	Immigration
10	Transportation
12	Law, Crime, and Family Issues
13	Social Welfare
14	Community Development and Housing Issues
15	Banking, Finance, and Domestic Commerce
16	Defense
17	Space, Science, Technology and Communications
18	Foreign Trade
19	International Affairs and Foreign Aid
20	Government Operations
21	Public Lands and Water Management

Note that Topic 11 is not defined.

III. Detecting Major Policy Agenda Sub-Topics in Political Tweets

In this section, we present the details of our proposed work for major policy agenda sub-topic detection. We begin by describing stage-1 tokenization, followed by major policy classification using CNN, stage-2 tokenization/lemmatization & pairwise word similarity calculation, tweet-similarity graph generation, tweet community detection and lastly, sub-topic extraction. (ref. to figure 1)

A. *Stage-1 tokenization*

For our experiment, we use public twitter APIs [13] to extract tweets from a pre-specified list of public twitter handles. The extracted tweets are stored in a SQL database for further processing. Though tweets are usually small (due to character count restrictions), they tend to be noisy. This requires that the noise be removed from the textual content before we proceed with further processing. A raw tweet can consist of a mixture of punctuations, hyphenations and abbreviations. To allow the CNN to train based on the meaningful content of a tweet, the pre-processing step filters out stop words, punctuations and removes IDs of other twitter users from the tweet.

B. *Major policy classification using CNN*

Prior to detecting policy agenda sub-topics, we would like to achieve a preliminary classification of all our tweets. This is achieved by classifying each of our tweets to a pre-determined set of domain-related topics (in our case, the major US legislative policy agendas). For this, we use the publicly available US policy agenda codebook [11], which provides an exhaustive list of policy agendas in context of the US political system. To provide the classifier with this trained data, some of the tweets were manually classified to one of the topics from the codebook [11] by students from Department of Political Science at Iowa state University. For our work, we simply reuse the tweets originally annotated as part of a previous research done in [10]. This work aimed at showcasing the effectiveness of CNN in detecting major policy agenda topics in political tweets published by US state representatives. The CNN classifier was trained using 10-fold cross validation and the entire dataset (consisting of all manually annotated tweets) were used as our test set. The work also showed that CNN as a classifier achieved better accuracy as compared to SVM (Support Vector Machine) based classifiers on our dataset.

As an overview, Policy Agendas Project [11] defines 20 major topics and 220 subtopics of policy agendas in a codebook [11] as presented in Table 1. By classifying tweets into major policy agenda topics, we aim to reduce overall load on community detection and topic extraction modules by clearly separating tweets targeting different classes of legislative policy.

We use the CNN classifier tool developed in [10] to classify the processed tweets to one of the 21 major policy agendas from the codebook. Although the classifier achieved a stable prediction accuracy, for a given test tweet, a trained classifier outputs only a single major policy agenda topic (the topic with the highest conditional probability). However, this result is prone to misclassification due to the inherent imperfections of trained CNN classifiers. To mitigate this scenario of considering only the most probable policy agenda, our method aims to extract multiple highly probable policy agendas that a given tweet might be targeting. To achieve this objective, we tweaked the code from the CNN classifier to fetch a list of all categories with probability values within the last quartile of the Gaussian distribution (constructed based on the conditional probability of all policy agenda classes). A new set of files were then generated, one for each major policy agenda, where a given tweet was inserted into all files which corresponded to policy classes from last quartile of our Gaussian distribution. At the end of this step, we had 21 files (each assigned to one of the 21 policy agenda topics) containing tweets which had a ”high” likelihood of targeting the given policy agenda.

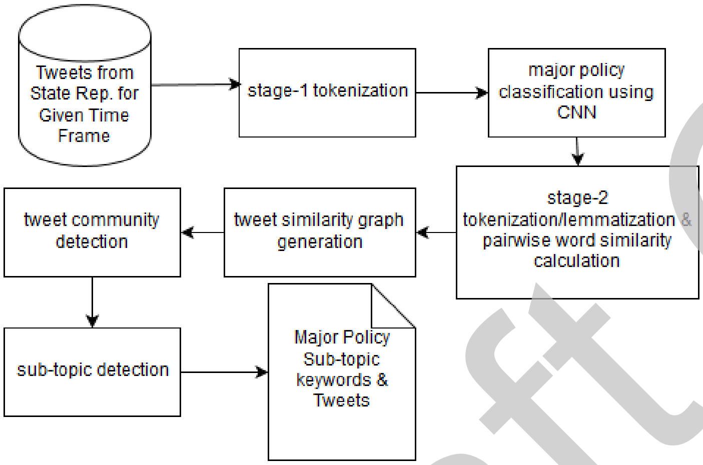
Tweets from State Rep. for Given Time Frame
stage-1 tokenization
major policy classification using CNN
stage-2 tokenization/lemmatization & pairwise word similarity calculation
tweet similarity graph generation
tweet community detection
sub-topic detection
Major Policy Sub-topic keywords & Tweets

Fig. 1. Sequence of stages involved in major policy sub-topic extraction

C. *Stage-2 tokenization/lemmatizaton & pairwise word similarity calculation*
The next set of steps are performed on each of the 21 files obtained post-classification.

- *Tokenization & Lemmatization*: We tokenize the tweet as before to remove noise. This is followed by a process called lemmatization, which reduces each individual word to its root word, making it easier to compare different words which derived from the same root. In our experiment, we use the libraries provided as part of the Stanford Lemmatizer [12]. This library helps with Parts-of-Speech (POS) detection, tokenization and lemmatization of policy tweets.
- *Pairwise word similarity*: In order to calculate text-based semantic similarity of 2 given tweets, we use the contextual-similarity metric to find how ”similar” each word from a tweet is to the words from another tweet. We use Java WordNet [14] to calculate this similarity score. This score is calculated on a scale of 0-1 (0 indicating extremely low similarity and 1 indicating a very high similarity score). As we are only interested in highly similar words, we record word pairs with similarity score of 0.7 or higher.

D. *Tweet-Similarity graph generation*
At this stage, we have, for a given policy agenda topic, 2 files. First, a file containing all tweets as assigned to the given policy agenda, represented in clean form. Secondly, we have a file containing all word-pairs with a high contextual- similarity score as calculated using WordNet[14]. We now present on how we use these 2 files to create our tweet-similarity graph, based on the cosine similarity metric for any tweet pair.

Since this experiment is focused on twitter, the cosine similarity calculation was modified to give importance to text items like hash-tags and proper nouns (countries, people, entities). This helps to lead to a more realistic score which is closer to human interpretation of a tweet (while comparing tweets, humans primarily look for nouns and hash-tags to identify the subject of the tweet). Parts-of-Speech like adverbs, adjectives, verbs, common nouns are given lower priority while calculating the cosine similarity score. For our experiment, we refer to this priority as ’boosting factor’ (higher the ’boosting factor’, higher the priority) which is assigned based on Parts-of-Speech. We assign a boosting factor of 0.2 to adverbs, adjectives, verbs, common nouns, a boosting factor of 1.0 to proper nouns and 1.3 to hash-tags. The factors are multiplied with the respective pairwise contextual-similarity scores as shown in eq. (1) to calculate the numerator part of our cosine similarity.

While we use WordNet [14] to calculate contextual-similarity for word pairs which are not hash-tags, we follow a different approach for calculating how ”similar” 2 separate hash-tags, which are a prime feature of twitter. Hash-tags serve as a very important tool to aggregate tweets talking about the same topic, even though they might be using the different expressions or words. However, WordNet [14] cannot implement the task of comparing hash-tags. We instead use a data-structure called UnionFind, which can be used to group together hash-tags based on our ’similarity’ metric, calculated

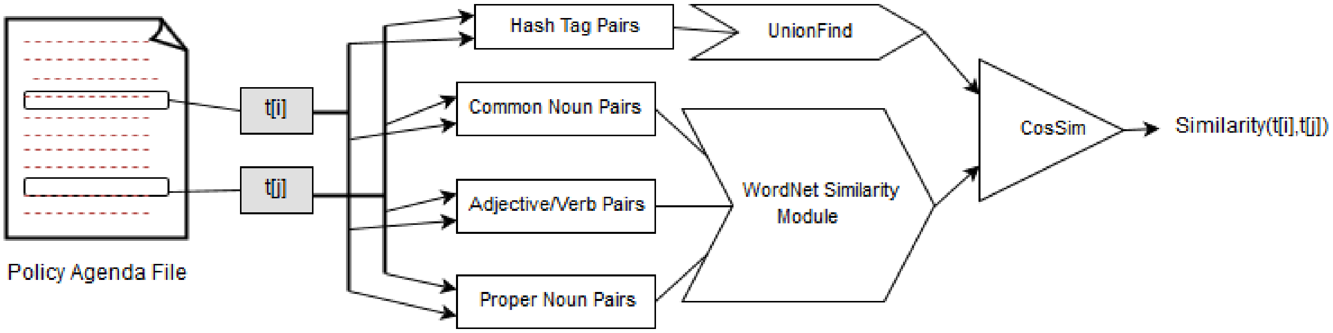

Fig. 2. Cosine Similarity calculation steps for any given tweet pairs t[i] & t[j]

as follows:

- Initially, we create a new group for each distinct hash-tag used in any tweet for the given policy topic.
- We iterator through all tweets in the policy topic file, finding pairs of hash-tags which occur together in the same tweet. On finding such pair, we merge together the 2 groups that each of the hash-tags belong to.
- This process continues till we reach the last tweet. In the end, we have a clusters of hash-tags, where each cluster contains hash-tags which are "contextually" similar based on their cooccurence in a tweet.

We now consider the equation used for calculating the cosine similarity of 2 tweets, namely t_i & t_j. In equations (1) to (5), the terms [*C, P, AV, H*] represent the set of common terms (common nouns, proper nouns, adjectives/verbs and hash-tags respectively) for given tweets t_i & t_j. The function *f()* depicts the contextual-similarity score for a given term pair, obtained using WordNet [14]. Functions *c(),p(),a(),h()* compute the contextual-similarity score sum of all possible pairs of common nouns, proper nouns, adjectives/verbs and hash-tags respectively for t_i & t_j. *n()* represents the number of term pairs for each given POS (Parts of speech) class for the tweets pair t_i & t_j. *Num()* computes the numerator of the cosine similarity metric, where *wt()* represents the boosting factor of the respective word construct.

$$c(t_i, t_j) = \sum_{1}^{n(C)} f(C) \tag{1}$$

$$p(t_i, t_j) = \sum_{1}^{n(P)} f(P) \tag{2}$$

$$a(t_i, t_j) = \sum_{1}^{n(AV)} f(AV) \tag{3}$$

$$h(t_i, t_j) = \sum_{1}^{n(H)} f(H) \tag{4}$$

$$\begin{split} Num(t_i, t_j) = wt(C).c(t_i, t_j) + wt(P)p(t_i, t_j) + \\ wt(AV)a(t_i, t_j) + wt(H)h(t_i, t_j) \end{split} \tag{5}$$

To calculate the denominator of our cosine similarity metric, we simply take the product of the L2/ Euclidean distance of the modified frequency vector. (frequency of each term is multiplied by the corresponding POS boosting factor to form the modified frequency vector). This is followed by computation of the cosine similarity score. Here, function *d()* represents denominator for cosine similarity and *cos()* represents the actual cosine similarity for tweets t_i and t_j

$$d(t_i, t_j) = L2(t_i) * L2(t_j) \tag{6}$$

$$cos(t_i, t_j) = n(t_i, t_j)/d(t_i, t_j) \tag{7}$$

Now that we have the cosine similarity scores for all pairs of tweets for a given policy agenda, the next step involves construct a tweet-similarity graph using this similarity score. We filter out all tweet-pair which have a cosine similarity score below a given threshold value (indicating low semantic similarity). We will explain in the next section on how we choose this threshold score. The graph is constructed based on the following guidelines:

- Each node represents a single tweet from the given policy agenda file.
- Each edge represents that the nodes/tweets connected by the edge have a cosine similarity score above the given threshold.
- The weight of an edge is the cosine similarity score of the corresponding nodes/tweets multiplied by a factor of 10 (ranging from 0-10).

E. *Tweet community detection*

We now use a suitable community detection algorithm to extract possibly several tweet communities within each of the 21 major policy agendas topics. The intuition here is that applying an appropriate community detection algorithm to the Tweet-Similarity graph would give us communities of tweets which in turn, are highly similar to each other and that such a community would be formed based on the tweets talking about a common policy agenda sub-topic. Hence, each community detected would consist of tweets primarily focusing on a single sub-topic within each major policy agenda topic.

Modularity in a network is designed to measure the strength of division of a network into modules. It defined as the fraction of the edges that fall within the given groups minus the expected fraction if edges were distributed at random. A high modularity score means more edges lie

within the module than you expect by chance.
Although this is rare but we also need to keep in mind that a single tweet might be talking about multiple topics at the same time. We choose to apply the Walktrap community detection [9] to tackle our 2 main objectives : Firstly, to detect communities in weighted graphs and secondly, to detect overlapping communities, if any. The general idea of Walktrap is that if you perform random walks on the graph, then the walks are more likely to stay within the same community because there are only a few edges that lead outside a given community. The result is a set of overlapping communities, each containing a set of tweets.

F. *Sub-Topic detection*

Once communities of tweets are known, the remaining step is to extract a set of keywords (hash-tags and regular words) which define each resulting community. To achieve this, we revisit the process of filtering out pairs of tweets with qualifying cosine similarities (based on threshold). For each tweet pair which qualifies, we recompute the word-pairs which contributed to the similarity score. The aim here, is to construct for each community, a rank-based structure of regular words (hash-tags excluded), where rank corresponds to contribution of a given word to the community. The boosting factor is used to calculate the contribution. Given below is the module *CommunityTopicExtractor*, which iterates through all tweets in a community and uses boosting factor to calculate individual word contribution to output keywords ranked based on contribution. We maintain a separate structure HashTagRank to maintain hash-tag contribution, except that when encountering pairs of hash-tags, UnionFind is used to ascertain if the hash-tags belong to the same group. If they do, we add their contribution to HashTagRank.

Once we cover all pairs of tweets in a given community, we separate ranked keywords into 2 categories, namely hash-tags and regular words (nouns,verbs,adjectives). For a chosen integer [K] (K remaining constant across all communities), we choose the top K regular tokens for each community, based on ranking. As hash-tags are the most important topic indicator for a given tweet, we consider all the hash-tags for our output, hence hash-tags are not filtered out. Lastly, even though hash-tags are a good indicator of the topic discussed in a single tweet, their contribution to a community depends on the percentage of tweets containing hash-tags. For this, we split the percentage scale into 4 parts to specify hash-tag contribution to each community, as follows:

Algorithm 1: CommunityTopicExtractor

```
input : community list for Policy Agenda Topic $P_i$
output: List<significant-keywords>

for community $\theta$ of $P_i$ do
    SigRank ($\theta$) = ranked mapping of word
      contribution for community $\theta$;
    TArr $\leftarrow$ list of tweets from $\theta$;
    TPair $\leftarrow$ all pairs of tweets in TArr;
    for tP in $TPair$ do
        cosScore $\leftarrow$ cosine similarity score of tP;
        if cosineSim(tP) $>$ threshold then
            tokList = list of all word-pairs in tP with
              high contextual-similarity ;
            while ($w_a$,$w_b$) in tokList do
                bFactor $\leftarrow$ boosting factor for $w_a$ & $w_b$;
                //Update the contribution of token $w_a$,$w_b$
                //in structure SigRank
                update(SigRank, $w_a$*bFactor*cosScore);
                update(SigRank, $w_b$*bFactor*cosScore);
            end
        end
    end
    print topKSorted(SigRank ($\theta$))
end
```

- 0≤percentage≤25 signifies LOW importance
- 25≤percentage≤50 signifies MODERATE importance
- 50≤percentage≤75 signifies HIGH importance
- 75≤percentage≤100 signifies VERY HIGH importance

IV. Results

We describe datasets, experimental design, and results in this section.

A. *Datasets*

For our experiment, we used the same dataset as created for major policy agenda classification in [10]. This dataset was created by collecting 308,601 tweets from 472 official accounts of the Senate, House and of individual senators and house representatives from eleven states in US, during the time period of 06/29/2008 to 11/29/2015. The tweets and meta-data were stored in tables using "MySQL Community Server 5.7", which is a relational database software. Tweets from Iowa and Nebraska were chosen for manual labeling of ground truth by two political science students, based on guidelines specified in the codebook [11] and the guidance of a political scientist.The labeling process assigned each tweet to one of the 21 policy agenda topics from table 1 (Note that topic 11 is not defined). There was also a topic '0', namely 'Mixed', assigned for tweets which addressed multiple policy agendas. Each tweet was labeled by only one student.
For our experiment, we considered tweets published only by state representatives from the states of Iowa and Nebraska.

B. *Experimental Design*

- For the CNN classifier, we set the word's vector length to be 300, the set of the windows sizes to be 3,4,5, and the choice of the channel as 'Static' (the dataset being static in size). The parameter representing vector length was set to 300 based on the length of the largest vector created to represent a single word in the entire dataset, which was 300 for our dataset. We used 10-fold cross validation for training our classifier.
- For each major policy agenda topic, we stored contextual-similarity scores of word-pairs, calculated using WordNet [14]. We only stored word-pairs with a contextual-similarity score $\geq$ 0.7, considering that we were only concerned with word-pairs which were highly similar to each other. This helped us discard

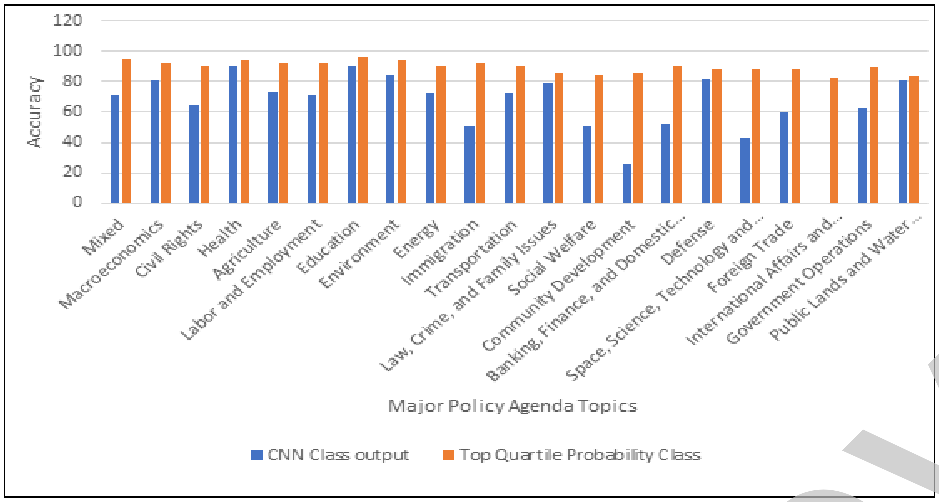

Fig. 3. Accuracy comparison of policy retention for tweets, for each major policy agenda

word-pairs which did not have a significant impact on the similarity score of 2 given tweets.

- For construction of Tweet-Similarity graph, we set the cosine similarity threshold based on gradient descent approach. The cosine similarity was set to a starting value of 0.8. Communities were detected using Walktrap community detection method and the resulting modularity score of the communities were noted. This cosine similarity value was then decremented by a value of 0.3, resulting in a new Tweet-Similarity graph. The steps of finding communities on this new graph and recording the modularity score was repeated. These steps were repeated till modularity score did not dip by more than 20% of its value from previous iteration, where each iteration consists of decrementing the cosine similarity threshold by 0.3 followed by calculating modularity score of resulting communities. This helped make sure that we didn't lose well-defined communities (reduction of modularity score) for a minuscule increase in data retention (arising from lower contextual-similarity threshold)).

- For Walktrap community detection algorithm, we set the parameter weighted to TRUE.

As discussed earlier in section III, traditionally for a given tweet, CNN classifier outputs the policy agenda with highest likelihood based on the model constructed from training data. If the size of training data is unequal across different classes, we get a trained classifier which is better trained to handle classes with more training samples as compared to classes with lower amounts of data. This increases the odds of the classifier assigning a given test data to classes with bigger training sample size. This increases the probability of misclassification. In our dataset, certain policy agendas have been assigned more tweets than other policy agenda topics. To mitigate this effect on our results, we use the probability distribution vector to determine list of major policy agenda topics based on the last quartile in the probability distribution values.

We start with comparison of accuracy results for the two cases, specifically (i) when taking only the policy agenda with the highest probability (as outputted by CNN classifier) and (ii) when considering policy agendas with probability value in the last quartile of the class probability distribution. Results from figure 3 show that for each major policy agenda topic, using output from case (ii) allows us to retain the true policy agenda for more number of tweets as compared to the method from case (i). This is due to the fact that for a given tweet, case (ii) keeps track of possibly multiple major policy agendas with a high probability value, rather than storing just a single policy as outputted by a CNN classifier, thus increasing the odds of retaining the true major policy agenda topic for every tweet. Although this increases the percentage of false positives, our main objective here is to prevent data loss due to misclassification.

- *Sub-Topic Ground Truth* : To establish the sub-topics discussed in each of the major policy agenda topics, we asked domain-aware participants to go through each file and create a list of major sub-topics discussed. The responses were then normalized to remove similar worded sub-topics and produce a clean list of sub-topics. A different set of domain-aware participants were then asked to classify each tweet to one of the given sub-topics presented in the "clean" sub-topic list. This process produced a list of major sub-topics discussed, ranked based on the number of tweets discussing each given sub-topic.

- The chart in figure 4(i) displays the different threshold values fixed for cosine similarity calculation and for community detection modularity scores. Lowering the cosine similarity threshold helps retain more weighted edges, however, a densely connected graph is more likely to give poorly defined communities, resulting in a lower modularity score. This threshold values for each major policy agenda were chosen so as to ensure minimum data loss by pushing cosine similarity cut-off as low as possible while maintaining a reasonably high modularity score (quality of communities detected).

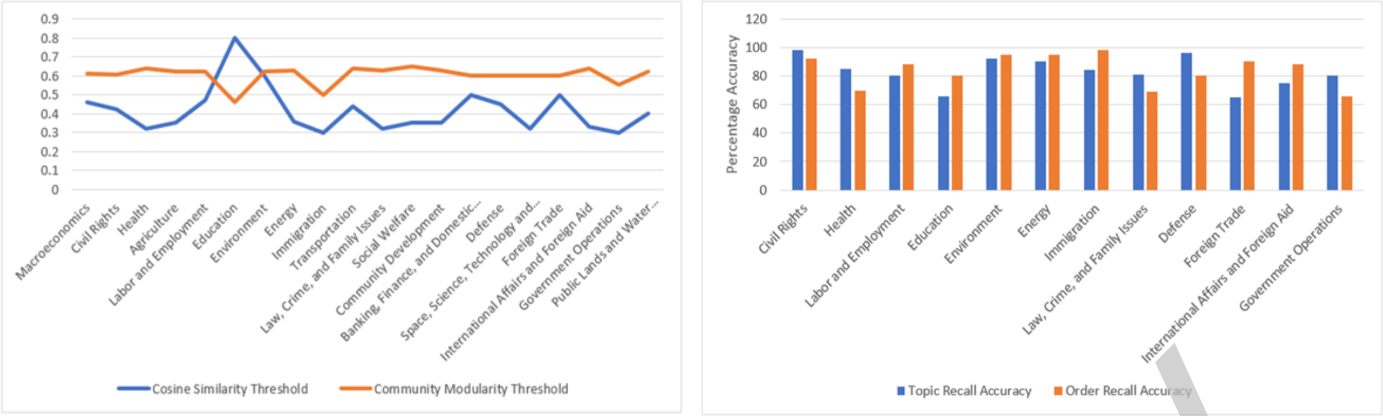

Fig. 4. (i) Threshold values used Tweet-Similarity Graph and community detection (ii) Accuracy comparison for Topic Recall and Order Recall

For sub-topic comparison, we filtered out some major policy where participants found it tough to define clear sub-topics being discussed based on tweets. For instance, participants lacked domain knowledge in Macroeconomics to be able to deduce topic based on tweets, hence, Macroeconomics was dropped from the experiment for sub-topic extraction. If you compare the x-axis from figure 4(i) & figure 4(ii), the latter has fewer major policy agenda topics listed.

To compare results of sub-topic keyword extraction, we compare sub-topics obtained from participant classification with the keywords outputted based on our proposed work. Comparison is done based on the following two aspects as defined below:

- *Topic Recall* : For our experiment, this is defined as the ratio (expressed as a percentage) of the number of major policy agenda sub-topics detected by our proposed work and the number of major policy agenda sub-topics listed out by domain-aware participants.
- *Order Recall* : To calculate order recall, we first sort the list of major policy agenda sub-topics outputted by our work, based on decreasing size of corresponding communities defined by each sub-topic. If we denote [C] as the ordered list of sub-topics outputted by our method and [M] as the ordered list of sub-topics as decided by manual annotation, then order recall is the percentage of adjacent pairs from [C] that do not violate sequence order in [M]. Our intuition is to penalize for every instance where the output sequence breaks off from the actual order. Mathematically, we calculate Order Recall as 100 - 100 * (mismatches / size(LIST(A))-1), output this percentage as Order Recall. If [C] = [2,4,1,3] and [M] = [1,2,3,4,5,6], we have adjacent pairs [4,1] which violate order in [M]. Hence, the order recall comes out to be (100 100*(1/3)) = 66.66% order recall. If [C] = [4,3,2,1], we have 3 mismatches, resulting in 0% order recall.

As shown in figure 4(ii), we got high accuracy values for both Topic Recall and Order Recall when compared with results from responses of domain-aware participants (values are in percentages).

We observed that for a given file, the accuracy in percentage value dropped as the number of tweets increased, eventually hitting a plateau. This resulted from two factors, namely (1) How participants defined sub-topics and (2) Threshold value chosen for cosine similarity and modularity. Considering factor (1), participants defined a new sub-topic if they found a fixed minimum number tweets discussing the given sub-topic. As the number of tweets increased, this fixed number became lower in terms of size as a percentage. Considering case (2), the lower the cosine similarity threshold was pushed, the more densely connected the tweet-similarity graph became and hence, the lower the modularity score of the resulting communities. This resulted in communities with tweets encompassing multiple sub-topics, resulting in some sub-topics getting masked due to relative significance of other sub-topics within the same community.

Based on these two factors, as the number of tweets in a file increased, moderately similar tweets belonging to smaller sub-topics got assigned to larger communities during community detection phase. However, this ensured that major sub-topics were preserved even if sub-topics with a very low percentage share got discarded.

Topic Recall accuracy varied based on major policy agenda topic, ranging from a low of 65% for ”Education” to a high of 98% for ”Civil Rights” & ”International Affairs”. As mentioned earlier, even though the accuracy on ”Education” was low, the file contained more than 3000 tweets and the accuracy accounted for loss of minor sub-topics only. For Order Recall, we touched a low of 67% for ”Government Operations” and a high of 97% for ”Civil Rights”. The low percentage order recall for ”Government Operations” was mainly due to extremely low number of sub-topics (in this case, 4), amplifying a single mismatch in Order Recall. Based on our results, we were able to extract major sub-topics being discussed consistently across all major policy agenda topics, while also maintaining the order of sub-topics.

V. Conclusion & Future Work

A. Conclusion

In this work , we propose a methodology to detect major policy-agenda sub-topics by using CNN classifier, cosine similarity metric and weighted community detection techniques. We demonstrate that across all major policy agendas, using our proposed method gave us a Topic Recall (percentage of sub-topics extracted) accuracy of 65% or higher and an Order Recall (maintaining order of sub-topics based on significance) accuracy of 67% or higher.

B. Future Work

In the current version of our work, we do not consider the temporal relation while calculating similarity between tweet pairs, which could be one possible future direction of our work. We also did not consider the influence of an account which published a given tweet i.e. a rank of a given twitter account based on number of twitter followers. This could allow us to create a graph with weights given to each tweet node based on the influence of the account which published it. Hence, another future work would be to create graph(s) with weighted nodes to consider twitter account influence. We currently applied our proposed work to political tweets, however, this could very well be extended to different application areas. One such area would be Journalism, where our method could be applied to extract important stories covered in a certain time period. Our method could also be applied to the corporate world, where its crucial to monitor mails and topics they address in the past.

References

[1] S. Phuvipadawat and T.Murata, Breaking news detection and tracking in twitter, in Proc.Web Intelligence and Intelligent Agent Technology, IEEE/WIC/ACM Int. Conf., 2010, vol. 3, pp. 120-123

[2] J. Sankaranarayanan, H. Samet, B. E. Teitler, M. D. Lieberman, and J. Sperling, twitter stand: News in tweets, in Proc. GIS: 17th ACM Int. Conf. Advances in Geographic Information Systems, New York, NY, USA, 2009, pp. 42-51.

[3] D. M. Blei, A. Y. Ng, and M. I. Jordan, Latent dirichlet allocation, J. Mach. Learn. Res., vol. 3, pp. 993-1022, Mar. 2003.

[4] D.M. Blei and J. D. Lafferty, Dynamic topic models, in Proc. ICML: 23rd Int. Conf. Machine Learning, New York, NY, USA, 2006, pp. 113-120, ACM.

[5] J. Yang and J. Leskovec, Patterns of temporal variation in online media, in Proc. WSDM: 4th ACM Int. Conf. Web Search and Data Mining, New York, NY, USA, 2011, pp. 177-186.

[6] H. Sayyadi, M. Hurst, and A.Maykov, Event detection and tracking in social streams, in ICWSM, E. Adar, M. Hurst, T. Finin, N. S. Glance, N. Nicolov, and B. L. Tseng, Eds. Palo Alto, CA, USA: AAAI Press, 2009.

[7] H. Sayyadi, M. Hurst, and A.Maykov, Event detection and tracking in social streams, in ICWSM, E. Adar, M. Hurst, T. Finin, N. S. Glance, N. Nicolov, and B. L. Tseng, Eds. Palo Alto, CA, USA: AAAI Press, 2009.

[8] M. Conover, B. Goncalvez, J. Ratkiewicz, A. Flammini and Filippo Menczer, ”Predicting the Political Alignment of twitter Users” in 2011 IEEE International Conference on Privacy, Security, Risk, and Trust, and IEEE International Conference on Social Computing

[9] P. Pons, M. Latapy, Computing communities in large networks using random walks in Journal of Graph Algorithms and Applications, vol. 10, no. 2, pp. 191-218 (2006)

[10] R. Li, W. Tavanapong, D. Peterson, J. Wong, ”Bigdata in Politics: Predicting Policy Agenda Topics in twitter”, Technical report submitted for publication

[11] Code book, Available: http://comparativeagendas.s3.amazonaws.com/codebookfiles /TopicsCodebook2014.pdf

[12] Stanford Corenlp, Available: http://stanfordnlp.github.io/CoreNLP/

[13] Twitter Data API URL, Available: https://dev.twitter.com/rest/public

[14] WordNet-A lexical database for English, Available: https://wordnet.princeton.edu/wordnet/