City of Boston Priorities Media Sentiment Analysis Report

Yancheng Liu, Haiqi Ma, Karantonis Georgios, Ganz Faiz Andrea, Lu Peiqing
CS 506 - Computational Tools for Data Science (Spring 2019)
BU Spark! Hariri Institute of Computing, Boston University

PROJECT SUMMARY

The objective of this project is to understand the media coverage (news and social media) and the public response to the to the legislative agenda of Martin J. Walsh, the 54th Mayor of Boston. This implies understanding which of the agenda's priorities were covered by mainstream media, in what quantity, the time of response of mainstream media when covering them with respect to the press release, and the public's general sentiment regarding such topics. In order to achieve these goals, the priorities of the mayor's agenda have been identified through the *City of Boston* online database press release and a comparative statistical analysis of the coverage of these topics by mainstream media was performed in combination with a sentiment analysis of the public's opinion. The analysis performed was able to reveal the main topics that are covered by the mayor's agenda, the amount of coverage each of these topics received by mainstream media, the most discussed topics on social media, the change in coverage over time of mainstream media and the sentiment for each topic expressed by both mainstream and social media. The rest of our report is formatted on the following way; in the "Methodology" section we present the technical specifications of our system and the models that were utilized, in the "Results" section we present the results we achieved along with they compare to our initial goals and finally in "Future Work" and "Conclusion" we present an overview of our system and discuss ways that can further improved its capabilities.

METHODOLOGY

After obtaining all of the datasets by scraping the different websites of each source, the analysis was performed in accordance with the priorities demanded by the client. The analysis work consisted of three parts. The first was the preprocessing of the data and the

identification of the agenda priorities using Latent Dirichlet Allocation (LDA). The second was the mapping of our data to the identified agenda topics and statistical analysis of the coverage of the identified agenda's priorities. Finally, the third part was the sentiment analysis of the public's opinion.

Data Preprocessing

The data from mainstream media websites and social media was scraped without the use of a filter that would classify whether it was pertinent to the mayor's agenda. Therefore, it still included many unrelated content and irrelevant data fields. Hence the data required initial preprocessing, which was done by performing the following 4 steps:

1) *Data filtering*

   In order to determine which of the collected articles were relevant to the mayor's agenda, a five-layer feed-forward neural network was implemented which was trained for 100 epochs using articles from *Boston Herald* and *Boston Press Releases* as the training set, achieving an accuracy of 99.83% in the test set. By evaluating the collected data, it was easily deduced that the whole Boston Press Releases corpus is relevant to the mayor's agenda, while for the rest of the collections, it can be noticed that the only corpus that contains useful labels for its articles is the one from *Boston Herald*. In order to make good use of it, the articles' categories contained in *Boston Herald* were manually split into three categories; relevant, irrelevant and neutral, where neural contains all the categories which relevance cannot be surely identified since they may contain both relevant and irrelevant articles. To create the training set, all the articles from only the relevant and irrelevant *Boston Herald* categories along with the articles from Boston Press Releases were concatenated. In order to obtain useful results from the neural network, it was necessary to make sure that the extracted dataset was of sufficient size, otherwise, the model would overfit, and would have been unbiased. The first constraint was easily satisfied since the training set consisted of about 100 thousand articles. Regarding the second constraint, it was noticed that the irrelevant articles dominated the dataset since they reported for approximately 80% of it. Although this could reflect the actual relevance distribution in the entire collection, it would still add bias to the network. The issue was tackled by performing sampling in the negative articles, in a way such that the irrelevant articles would constitute approximately 60% of the training set. Each of our model's layers consisted of 200 internal nodes and was activated using the ReLU function (at first ELU was implemented but the accuracy ended up being a bit lower) and for the final

trained model 23 epochs were selected, since after that the network started overfitting the data.

Regarding the social media dataset, keyword matching was used to filter out irrelevant data, in order to insure that the final data set is related to the mayor. The keywords include the words “mayor”, “City of Boston”, “City Council” and the names of all the departments of the city of Boston. After further examination of the filtered data, keywords that may lead to keeping unrelated contents like “schools”, “libraries”, “retirement” and “landmarks” were removed from the keywords list. Finally, a new standard was added for the filter, which was that the body text should include at least one part or town of the Boston area. With the above filtering technique the results maintained about 3% of the original Twitter dataset.

2) *Removing stopwords, invalid characters, and digits*

The data was then cleaned using Python’s NLTK package. First, all the words in the documents were converted to lowercase and WordPunctTokenizer was used to tokenize them. Then, stopwords, invalid characters, digits and punctuation marks were removed altogether. The remaining tokens were lemmatized and each document was encoded into one embedding using the doc2vec vectorizer.

3) *Stemming or Lemmatization*

“Stemming and lemmatization are both ways for regularizing words by reducing them to subwords, but they follow significantly different intuitions. Stemming usually refers to a crude heuristic process that chops off the ends of words in the hope of achieving this goal correctly most of the time, and often includes the removal of derivational affixes. Lemmatization usually refers to doing things properly with the use of vocabulary and morphological analysis of words, normally aiming to remove inflectional endings only and to return the base or dictionary form of a word, which is known as the lemma” [Manning et. al, Introduction to Information Retrieval]. Since, as mentioned at the beginning of this paragraph, both of these methods are regularization techniques, applying both of them can prove to be harmful and thus, in the preprocessing, only lemmatization was performed.

4) *Feature Extraction*

To extract some efficient feature matrix for the following work, 2 methods were tried to extract features from text data. The first one is TF-IDF, which uses term frequency–inverse document frequency to reflect how important a word is to a document in the collection or corpus.

Although TF-IDF has been used for many years in Natural Language Processing, nowadays mode advanced models have been developed, rendering it obsolete. The current state of the art is Bidirectional Encoder Representations from Transformers (BERT), the stunning detail of which is that it does not rely on complex deep models but on multiple levels of self-attention and simple feed-forward neural networks. The problem with BERT, like any bidirectional language modeling tool (such as ELMO), is that in order to capture dependencies from both seen and unseen words it has very high requirements in memory. This is the reason why BERT could not be utilized for the modeling of the corpus and instead, doc2vec was chosen, which is based on the well known word2vec. Building on top of word2vec, doc2vec creates a single embedding for each document, which is able to capture the same semantic properties as word2vec. This provides a useful tool for grouping different documents since, with this model, similar documents tend to have similar angles in the defined vector space. Finally, doc2vec ws trained only on the articles from *The Boston Herald* and Boston Press Releases. The intuition behind this is that since only these sources were used for the filtering of irrelevant documents, and thus they need to define the space in which all the other documents will be projected, allowing for better filtering and classification.

Priorities Identification

Several techniques were tested for the mapping of the collected articles to topics from the mayor's agenda. It was decided to take advantage of the fact that the extracted articles from Boston Press Releases contain agenda topics, in the "published_by" feature, by training a classifier to predict the articles of the corresponding topics.

In order for the classifier to perform correctly, it had to be asserted that the training data were sufficient and unbiased. Since none of these constraints were satisfied, some preprocessing was required. Regarding the bias constraint, it turned out that the articles published by the mayor's office were dominating the dataset since they corresponded to approximately 40% of the dataset. Taking also into consideration the fact that there was a total of over 40 target classes, the decision, to completely drop from the training set the articles published by the mayor's office, was made. Also, articles from categories that contained in total of less than 10 documents were dropped as well. Regarding the size of the dataset, it ended up containing a bit over 2000 articles, which made it clear that trying to fit a neural network in such a dataset would lead to overfitting. In order to test this assumption, the data were fed to a feed-forward

neural network and it was noticed that even when using a single hidden layer the network overfit to its input data.

Another option that was examined was the use of an SVM model, which could potentially perform better than a neural network in such a small dataset. Unfortunately, though, our SVM model ended up achieving an accuracy of only 52%.

Since the examined supervised methods ended up yielding poor results, an unsupervised one was also tested. The intuition was that a clustering algorithm could potentially group together similar articles since our doc2vec embeddings are able to capture semantic similarities in the vector space, but the results, for k-Means, were again suboptimal. One possible explanation for this is that, as mentioned in the previous section, the metric for the semantic similarity is the angle of the vectors instead of their position in the embedding space. This means that the k-Means algorithm should be able to group articles based on their distance in the vector space and not based on their angle.

Based on the findings from the previous experiments, we tried to classify the extracted articles by exploiting their semantic similarities in the vector space. The documents corresponding to each of the target classes from Boston Press Releases we concatenated to one resulting in a single large topic document for each agenda topic, which was vectorized using doc2vec. Each of the articles, of the remaining mainstream media, was classified to a specific agenda topic by identifying the topic document with the minimum distance from it. Unfortunately, the high overlap between the different classes was again observed.

The final approach for the identification of the priorities was to perform Latent Dirichlet Allocation (LDA) or Non-negative Matrix Factorization (NMF) in the *City of Boston* press release; due to more clear and easily interpretable results the latter was selected. With this technique, 10 topics which are assumed to be representative of key topics covered by the mayor's agenda, were obtained along with a set of descriptive words for each topic. The classification of each article to a specific agenda topic was done using the majority voting scheme, meaning that each article was assigned to the topic which had its descriptive words as the highest overlap with the words contained in the article.

Sentiment Analysis

For the sentiment analysis, three methods were chosen to analyze the polarity of the data. The first method is based on the pre-trained, python library, TextBlob and a dictionary-based

method from Vader in NLTK. The second is based on the LSTM model implemented using Keras. The third and final sentiment analysis classifier chosen was Gluon NLP.

1) *TextBlob and Vader*

   TextBlob is a python library for processing text data. It provides a simple API for diving into common natural language processing, and it also supports the sentiment analyzer. The sentiment analyzer of choice was NaiveBayesAnalyzer (PatternAnalyzer is the default). The analyzer accepts a string as input and will return a named tuple, including polarity and subjectivity. For the purpose of this project, information about the subjectivity of the text was not needed, so only the polarity was used as the standard to decide whether the sentiment of the text was either positive or negative.

   A third type of polarity was added: neutral. The reason behind adding a new classification is because the sentiment analysis is performed over mainstream media articles data, which might not show any preference or bias, but might instead try to give comments more neutral in nature. The results also show that a large portion of data in the mainstream media is neutral. This is solid proof of the necessity of this third class. The thresholds to divide between positive, neutral, and negative are the following: greater than 0.15 for positive, in between 0 and 0.15 for neutral, and negative otherwise. As for the labels, 1 was used to represent positive, 0 to represent neutral, and -1 to represent negative.

   Vader was used to apply sentiment analysis in the project. The Vader sentiment analysis tool is a lexicon and rule-based sentiment analysis tool, embedded in the NLTK library, and is based on a paper written in 2014. The sentiment dictionary of Vader is based on the manual annotation. It includes more than 7000 words with sentimental value. The scale of sentiment intensity is between -4 and +4. The larger the value is, the more positive this word will be. The result returned from Vader includes 3 values, which is the possibility that this text is positive, neutral or negative.

2) *Long short-term memory (LSTM) model (Twitter)*

   LSTM networks are a variation of Recurrent Neural Networks(RNN) and for the past years, they have been producing state of the art results in a plethora of Natural Language Processing tasks. The intuition behind RNNs is that due to their architecture, they should be able to capture long term dependencies on their input data. In practise this architecture has proven to fall short in capturing such dependencies due to the issue of vanishing/exploding gradients. To tackle this issue,

two other variations of the RNN architecture have been proposed; the Gated Recurrent Units and the Long Short-Term Memory networks, both of which try to overcome the problem of vanishing/exploding units by implementing gates in the cells of the networks. Although none of these architectures manages to completely overcome the problem of vanishing/exploding gradients, they have been able to define the state of the art in various Natural Language Process tasks and LSTMs have been established as the go-to architecture (it is worth mentioning that the last couple of years there has been a shift towards Google's Transformer model).

The model was trained based on Sentiment 140 dataset with 1.6 million tweets on Kaggle and predicted the attitude of the Boston area mayor related twitter data in the past five years. Based on the 1.6 million tweets, we extract the text part of the dataset and remove all invalid components such as links, punctuation and stop words from the original data.

The doc2vec model was trained with the processed data and get 300-dimension word vectors. Tokenizer was used to vectorize the original text and use pad_sequences to convert the resulting sequence into a two-dimensional numpy array. With word vectors and two-dimensional numpy array, an embedding matrix can be created, which is used to pre-trained word vectors in the Embedding layer of Keras.

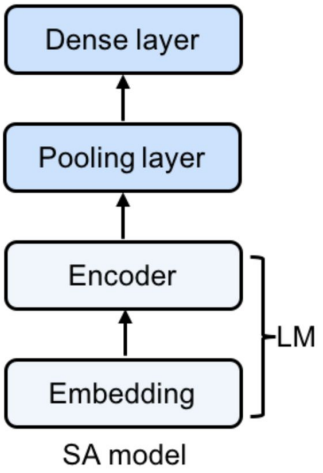
A two-layer LSTM model was used to achieve the accuracy of 0.7964 on the training set and the accuracy of 0.7949 on the verification set. This model is used to predict sentiment polarity, which is similar to the previous way of TextBlob and Vader. The prediction criteria is determined by adjusting different thresholds.

The model summary is shown as follows:

```
_________________________________Model Summary_________________________________
_________________________________________________________________________
Layer (type)                 Output Shape              Param #
=========================================================================
embedding_1 (Embedding)      (None, 300, 300)          74706600
_________________________________________________________________________
lstm_1 (LSTM)                (None, 300, 128)          219648
_________________________________________________________________________
lstm_2 (LSTM)                (None, 128)               131584
_________________________________________________________________________
dense_1 (Dense)              (None, 1)                 129
=========================================================================
Total params: 75,057,961
Trainable params: 351,361
Non-trainable params: 74,706,600
_________________________________________________________________________
Train on 1151999 samples, validate on 128000 samples
```

3) Gluon NLP

The Gluon NLP is a versatile toolkit that utilizes state of the art Natural Language Processing models and can be used in a variety of Natural Language Processing tasks. For the purposes of this project, a Convolutional Neural Network (CNN) based model was built on top of a pre-trained two layer LSTM language model. In order to be able to perform sentiment analysis, the whole model is trained on the IMDB dataset achieving 86% test accuracy. It is worth mentioning that training on the IMDB dataset will allow the model to generalize quite well on the mainstream media dataset, since, although the IMDB movie reviews may contain more easily sentimentality classified words, the total layout of both the reviews and the news articles can be quite similar.



Gluon NLP Sentiment Analysis model Architecture

Statistical Analysis

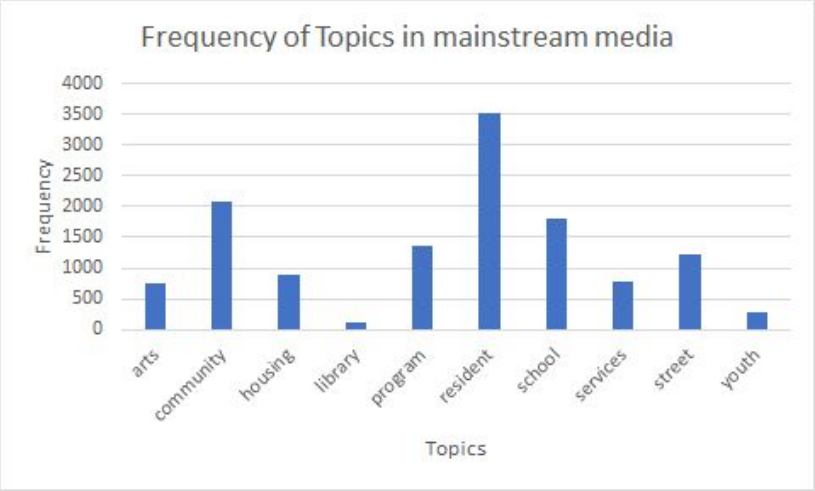
In the statistical analysis, we did some statistical work to our raw data, including separate data set into topic-related and count the number of releases with regard to each topic everyday to see if the press release is covered by mainstream media and social media and the change of coverage over time.

In order to answer the priority questions, we have to apply statistical analysis on our data set, and use the statistical data to answer the questions. We applied statistical analysis on mainstream and social media data set. And we will see the strategies of how to analyze the data using statistical methods in the following chapter, while we answer the priority question by analyzing the statistical data.

RESULTS

Priority #1:

1. **What key agenda topics are getting articles (i.e. covered) by mainstream media**
   We answered this question by topic words matching in the original dataset. There are ten selected topics: arts, community, housing, library, program, resident, school, services, street, youth, which are all chosen according to their frequency, representativeness and generality.



   From the chart above, the topics [Resident, Community, School] get the most coverage and attention in mainstream media, which means

2. **Did press releases or key events generate media coverage or awareness, i.e. look at timing and exact phrases matched from press releases?**
   For each topic, process the original dataset as a format of date and number of releases. Then we can see the time period that each topic invokes the public's attention.

   To solve this problem, we use the government department name list. At first, we found out those news containing a department name in press release news which often means that it contains a key event in the news of certain date. By manually searching for the key events, we know what it is. Then, we search the mainstream media news for the same events to decide whether the coverage of key event or agenda in the press

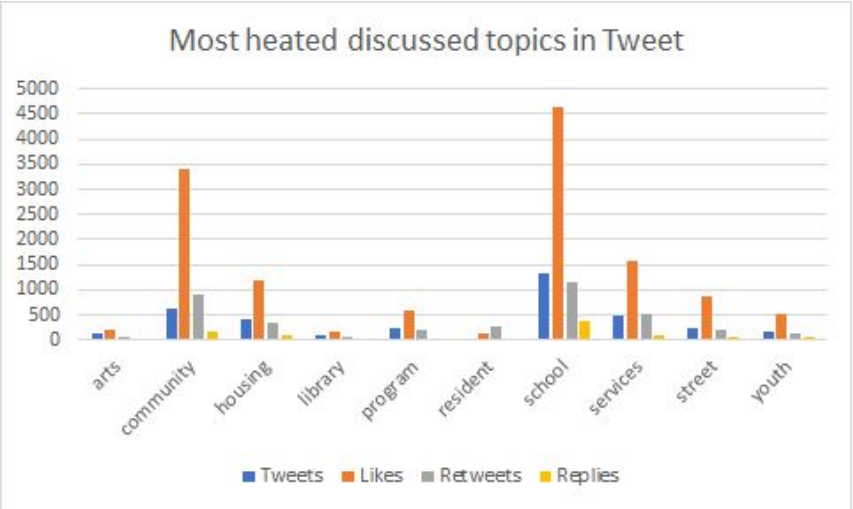
release generates media coverage or awareness. The phrase I chose is 3 days, considering the timeliness of news. It means that we have to check if the mainstream media has the same coverage in three days after the coverage in press release. As a result, we found that the answer is yes. For instance:

- On May 29, 2019, the press release covered the news of the mayor promising to provide housing services for those homeless veterans. In three days, the *Boston Globe* has covered the same topic. The news is about Veterans Services.
- On October 4, 2019, the press release covered that the government is going to propose regulations that aimed at addressing vaping and tobacco use among youth. Several mainstream media have related coverages in three days.
- On May 7, 2019, the press release covered that the government uses several measures to promote independent living for adults with disabilities.

There are also a lot of similar examples which sufficiently prove that press releases or key events generate media coverage or awareness.

3. **What key agenda topics are getting the most attention on social media (retweets, shares, likes, etc.)**

To solve this problem, we search for news items from Tweet which contain those ten topics by filter crawler. In this way, we get 10 data files, which correspond to ten topic words. Then, we count how many tweets, and the number of retweets, likes, replies they get in every data file. The statistic result is as follows.



As is shown in the chart, topics [school, community, housing, services] get the most attention and coverage.

4. **Which social platform has the highest engagement on key agenda topics?**
   Unfortunately, due to Facebook's privacy laws, we were not able to acquire data from it. Hence, the only social platform we were able to acquire data from is Twitter and no comparison can be made to other.

5. **Which hashtags are most popular on related topics**
   we clean the 'hashtag' column data simply using the replace and split method in string package and get a list of hashtags. Then, we count and rank the hashtags by the occurrences of them on related topics.

   For the social media part, for example, the most frequent hashtags on related topics are as follows:

BostonPublicSchool	97
Affordablehousing	69
YouthWork	92
teaching	38
kids	32
Youthwork	32
StudentOpportunityAct	19
jfklibrary/Bostonpuliclibrary	21
bostonmarathon	33
jobs/interns	32

From the table above we know that people on Twitter are more concerned about education, police, children/youth, housing and employment.

For the social media part, for example, the most frequent hashtags on related topics are as follows:

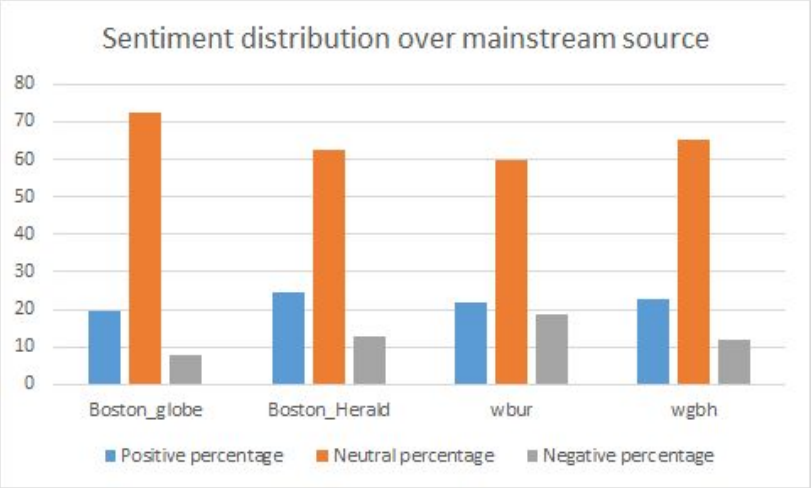
Boston Public School	228
Boston City Council	69
Boston Police Department	114
MTBA	114
Boston Redevelopment Authority	54
Youthwork	62

Boston Fire Department	30
Public Health Commission	24
Boston Public Library	33
Massachusetts Gaming Commission	23

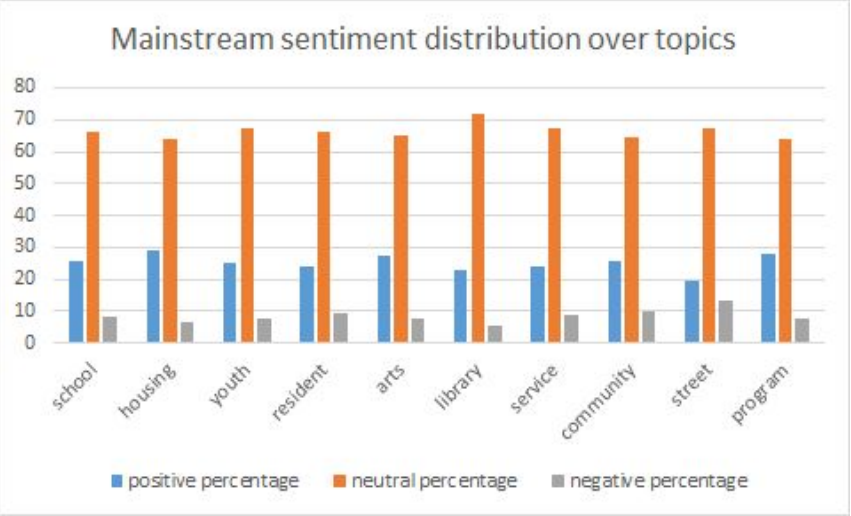
From the table above we know that, mainstream media pay more coverage and attention in government apartments.

Priority #2:

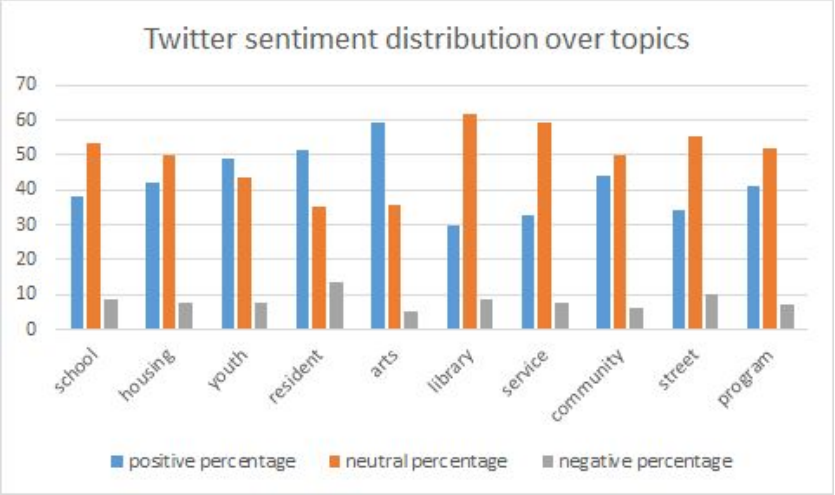
6. **What is the sentiment of this coverage i.e. positive or negative**



The graph above is the sentiment distribution over different mainstream sources. From this graph, we can see that most of the news about the mayor's agenda from mainstream media is positive or neutral. Only a few percent of news is negative. And according to the different data sources, the *Boston Globe* has the lowest percentage on the negative polarity and the highest percentage on the neutral polarity, while the *WBUR* has the highest percentage on the negative polarity and the lowest percentage on the neutral polarity.



The graph above is the sentiment polarity distribution of mainstream media over topics. As the distribution over data source, most of the news appear positive or neutral polarity. The topic has the highest percentage of negative news is street, which has a 13.05% of negative news. All topics have more than 85% of positive or neutral news. It proves that most of the actions and strategies of *City of Boston* are accepted and praised by the mainstream media.



And from Twitter data set, we can see the sentiment distribution over different topics in the graph above. Most of the tweets appear positive or neutral polarity on the mayor's agenda. Among 10 topics in total, the topic which has the highest percentage of negative tweets is the resident, street also has a relatively high percentage of negative tweets. Surprisingly, the topic arts get the most percentage of positive tweets, which we thought is the result of *City of Boston* paying attention on the development of arts.

In general, the public has a positive attention to the *City of Boston* and the mayor's agenda.

7. **Are there any patterns in press releases that lead to higher or lower public response?**
   We analyze the monthly distribution of mainstream media data on different topics, and try to find some patterns that invoke higher public response. But we find that the most important factor that influences the level of public response is the amount of press release that related to the topics. If the press release of *City of Boston* pays more attention on a specific field, the mainstream media will do it, too.

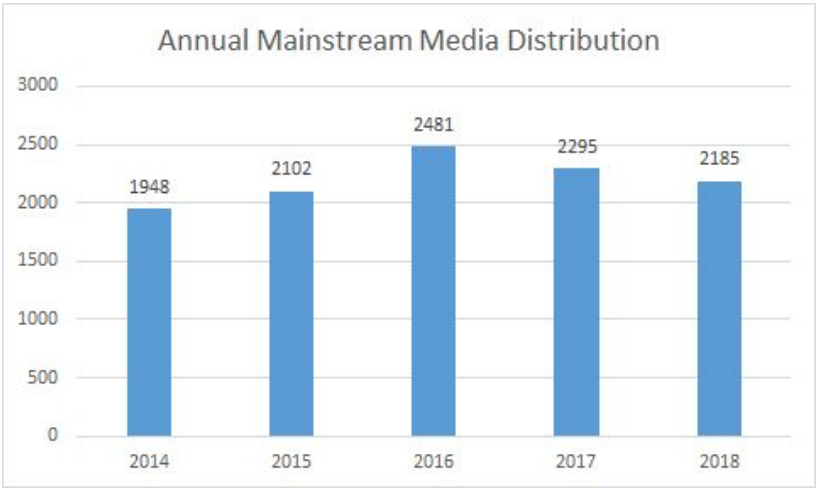
   For example, The amount of mainstream media on topic resident reach its peak, 83, in Jan. 2017. And we find that in the press release in Jan. 2017, 29 releases in total, 22 releases mentioned resident. That's why mainstream media pays the most attention to it.

   And in Jun. 2015, the topic school has a large amount of mainstream media news, 58, while in Jul. 2015, the amount of topic school decreases to 11. We looked back to the press releases of *City of Boston* in those time periods. And the major factor of this result is the content of press releases. In Jun. 2015, the topic school was mentioned 9 times, and 5 releases was about school, while in Jul. 2015, only 1 release was about school. And we can see the decrease of public response is mainly due to the decrease of attention of the *City of Boston*.

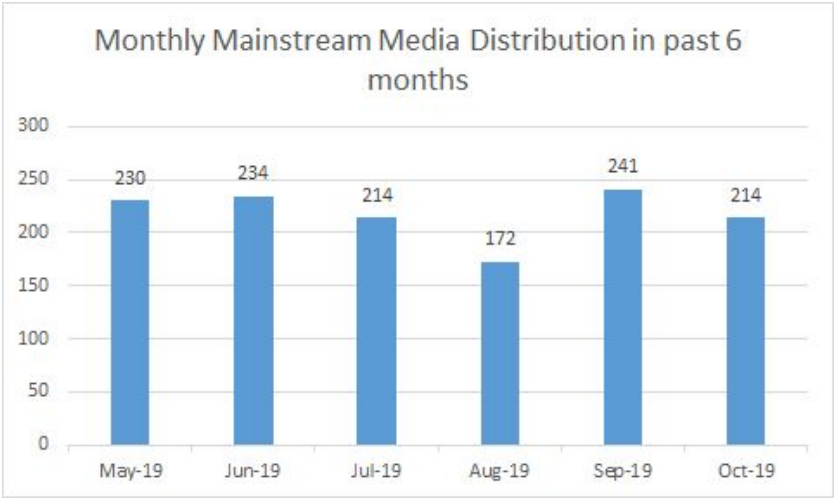
8. **Are there any agenda topics not commonly covered by the mayor's political team that are widely talked about in social media?**
   Because of the limit of our Twitter data set and scraper, we cannot have more data about *City of Boston*'s agenda, and we only have data related to specific topics. So right now we are unable to answer this question, but we can answer it in the future work once we can get more data about the mayor's agenda.

Other Questions:

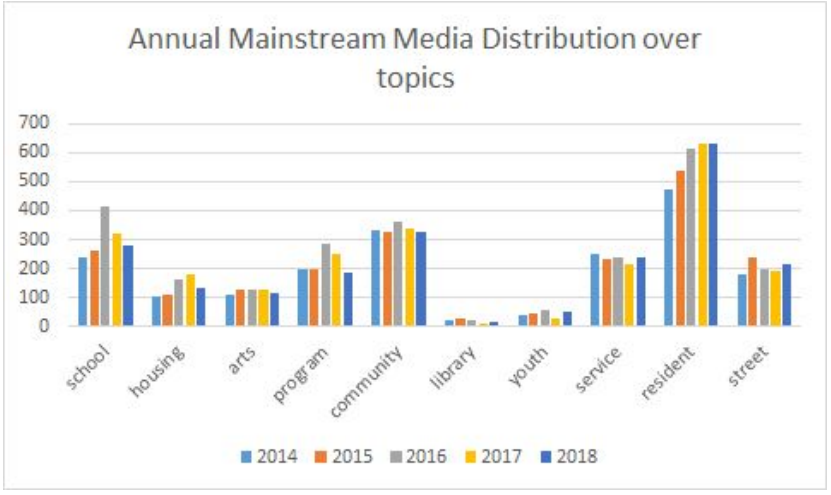
9. **How has the public response to the mayor's agenda changed over time annually and over the last 6 months by month**



The graph above is the annual distribution of mainstream media data. As we can see, the year with the most amount of news is 2016. And mainstream media paid more attention on *City of Boston* in recent years.



Over the past 6 months, the amount of news in mainstream media is relatively stable. The monthly amount of news is higher than 200 except August.



The graph above is the annual changes of mainstream media on topics. As we can see, mainstream media's attention on resident increase annually. And the amount of school, program and community related news are highest in 2016, then return to the normal level. The library and youth get relatively little attention of mainstream media, and even less in recent 2 years.

10. **Where (geographically) are key pieces of agenda not being covered?**
    For the time being, this question cannot be answered with the data acquired, as there is no information about the geographical distribution in any of the mainstream media data. Regarding the Twitter data, we tried to scrape them using the zip code but the result was not sufficient to answer this question.

FUTURE WORK

In this project, various methods to analyze the data from press release, mainstream media and social media were examined. Our work could be further improved by collecting more labelled data fine tuning two of our system's submodules; the priorities' identification and the sentiment analysis.

Regarding the priorities' identification, the need for a concrete classification scheme is imperative. We believe that classification based on the documents' could give better results but additional tests and finetuning are required. Regarding the sentiment analysis engine, even though our best working model manages to achieve a high accuracy of 86%, it could be modified to score up to 93% by implementing the current state of the art model.

Finally, one of the biggest constraints we had to overcome was the lack of labelled data. We argue that finding sources, such as websites, from which such data could be extracted, could not only give a significant boost to our system's performance, but also could open new possibilities in the models that could be utilized.

CONCLUSION

For the purposes of this project, we created a collection of data by scraping the following four online media outlets; *Boston Press Releases*, *The Boston Globe*, *The Boston Herald*, *WBUR*, and e*WGBH*, along with data collected using the Twitter API. The collected data expand from 2014 to this day and contain all the tweets and the articles, along with their metadata, detected in this time period. The topics of the mayor's agenda were determined by applying Non-negative Matrix Factorization on the Boston Press Releases dataset.

The mainstream media filtering was performed by training a 5 layer feed forward neural network on a sample of the original dataset, achieving an accuracy of 99.83%. The filtered articles were assigned to their corresponding agenda topics using a majority voting based classifier and finally, the classification results were passed to the Gluon NLP sentiment analysis system, that achieved an 86% accuracy.

The data collected from Twitter were filtered using a keyword matching technique and were assigned to their corresponding agenda topics using a majority voting based classifier. The sentiment analysis was performed by a 2 layer LSTM network, that achieved an accuracy of 79.49%.

Finally, the same statistical analysis procedure was performed to all the data allowing us to answer the majority of the priority questions.