Describe the City You Live in
Understand the City with Social Sensing

Zhuqi Li, Dan Tasse, and Jason Hong

Abstract

Sensing the dynamics (e.g. noise and traffic) of a city on a large scale has traditionally been a challenging endeavor, often requiring deploying a large amount of sensors throughout the city, usually costing a great deal of money. To address this difficulty, we introduce, social sensing, an infrastructure-free sensing model and research methodology for studying the dynamics of a city on a large scale based on the social media its residents generate. We apply this new methodology to data from approximately 1.8 million

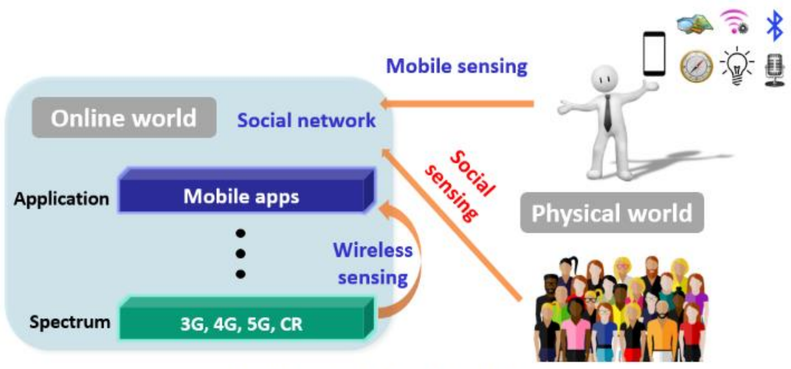
check-ins collected from users of Twitter in Pittsburgh. We build a hybrid model to understand the residents' opinion toward different metrics in the city. We use experiments to validate the performance of our system and visualize the result in an interactive way.

Overview

Social sensing is a sensing architecture that views every individual with a smartphone as a sensor. People tend to post message in the social network when they come across some events in the city. So by mining the social network, we can get the people's attitude toward the city as well as detect the events happened in the city.

There are four key components in our social sensing system,

- **Topic selection** helps us find the topics that people interested in about the city;
- **Keywords selection** helps us generate a bag of topic related keywords;
- **Tweet-topic matching** helps us determine whether the tweet is related to a specific topic;
- **Sentiment analysis** helps us determine people's attitude toward a certain topic.

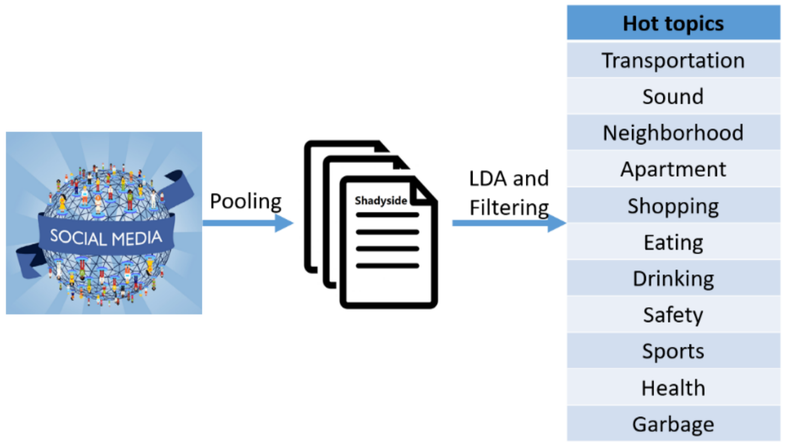


Architecture for Social Sensing

Topic selection

The topic selection part focuses on selecting the topics that people are really interested in about the city. The topic selection involves three steps

- Use pooling method to aggregate the tweets in the same neighborhood to form a virtual "document";
- Run LDA algorithm with different virtual documents and get a set of topics;
- Filter the urban-related topics among the general topics.



Working flow for topic selection

Keywords selection

The keywords selection is a part that generates a topic related bag of keywords to a certain topic. The keyword selection involves two steps.

- Train the word2vec model and get the vector representation of every word.
- Filter a bag of keywords iteratively for each topic based on the cosine similarity to the topic.

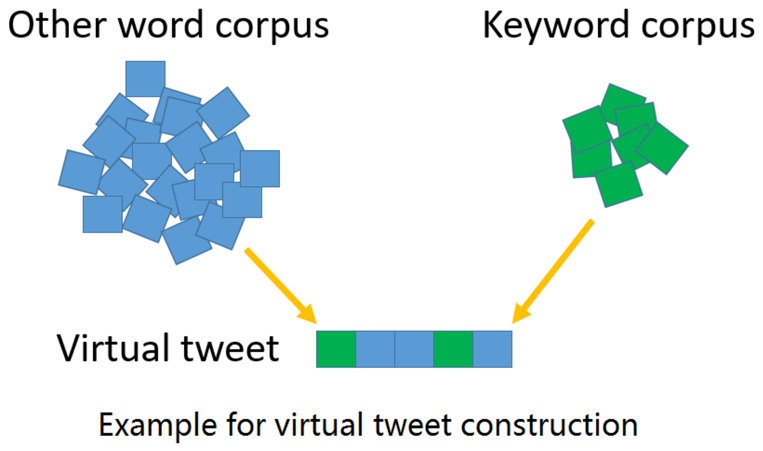
Iteration	Keywords
0	traffic, transport, bus
1	intersection, shuttle, cab, pedestrian, transit, tunnel, authority, vehicle, passenger, plow
2	pedestrian, port, lyft, passenger, uber, transit, highway, megabus, vehicle, suv
3	pedestrian, passenger, taxi, 71d, suv, transit, tractor, bumper, truckload, motorcycle
4	passenger, pedestrian, truckload, hazmat, tractor, suv, req, bumper, transit, motorcycle

Table 1: A demo for keywords selection.

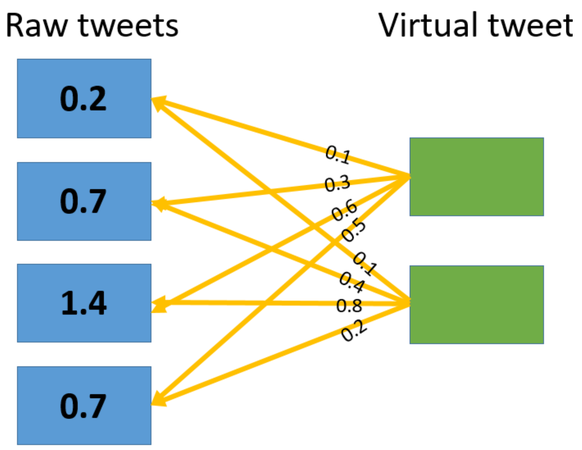
Tweet topic matching

The tweet topic matching is a part that determines whether a tweet is related to a specific topic. The tweet topic matching part takes four steps.

- Construct some virtual tweets by sampling the words with high similarity to the topic.
- Train the Doc2vec model and get the vector representation of both raw tweets and virtual tweets
- Let the virtual tweet vote for the raw tweets by their cosine similarity
- Set a threshold for the raw tweets to determine whether it related to the topic.



Example for virtual tweet construction



Example for the voting process

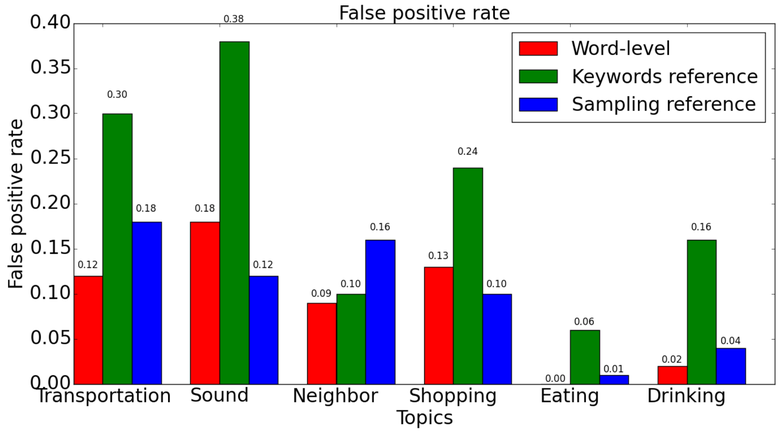
Sentiment analysis

We use the python NLTK to analyze the compound emotion of each tweet. For each tweet, the values of emotion ranges from -1 to 1, where 1 stands for the most positive emotion, -1 stands for the most negative emotion and 0 stands for the neutral emotion.

Experiment

- Dataset: 1.8 millions geo-tagged tweets in Pittsburgh with text, coordinates, emoji, time, user, hashtag, user profile.
- Results:

1. Topic matching result: we compare the false positive rate of our tweet topic matching algorithm to the baseline algorithm. And show the comparison result in the figure.



2. Interactive result: scan the QR code to see our interactive result.