MINING KNOWLEDGE FROM SOCIAL MEDIA DATA DURING CRISIS EVENTS

1 Introduction

The goal of this project, **from the web mining view**, is twofold:

1) Describe and analyse the social web graph based on its content, usage and structure facets;

2) Mine relevant knowledge from the social web graph for decision-making.

2 Data Description

The dataset contains real-world posts of the Twitter social platform (https://www.twitter.com) posted during past crisis events (floods, fires, tornado, earthquake, etc.). The dataset has been released by the NIST TREC Incident Stream Initiative organizers (https://www.dcs.gla.ac.uk/~richardm/TREC_IS/2020/task.html) whose main objective is to foster research and academia to automatically process social media streams during emergency situations, categorizing information and help requests made by citizen for emergency operators. The dataset mainly contains these high-level data, to make emergency possible based on data processing:

- **Events:** correspond to occurred crisis events (e.g., "*2012 Guatemala earthquake*", "*2013 Alberta floods*").
- **Event Type/Topic**: corresponds to an event category (e.g., "*flood", "earthquake", "typhoon", "bombing"*).
- **User**: an active twitterer on the platform during the event.
- **Tweet**: corresponds to a post a **user** emitted during a crisis event. Each tweet belongs to one **event**. Each tweet could be a **retweet** or a **reply** to another tweet; a tweet could **mention** a user, could contain a **hashtag** (e.g., "*nswfires"*, "*abflood"*) which is a keyword that relates to the event. A tweet includes a piece of text which corresponds to its main textual content (e.g., *"Alberta leaders give High Level wildfire update. Background here: https://t.co/qtkLlydctt https://t.co/9TkJi7jfQ0").*
- **Tweet Priority**: a human-labelled information about the level of criticality of a tweet to indicate the level of emergency. We distinguish between 4 levels of priority: *Low, Medium, High and Critical*.
- **Tweet High-Level Category:** a human-labelled information about the information types that might be needed by emergency response officers across a range of disasters, e.g., *Advice, News, Volunteer, search rescue,* etc.

3 Data Files

The original raw data is provided in the **data** sub-repository of the DATASET repository of the Project Moodle space. Such raw data requires cleaning steps such as removing spams, duplicates, undefined users and posts.

You can find in the **database** sub-repository the cleaned data divided in 3 sub-repositories including files in the JSON format:

- **Nodes**: mainly includes ***Event***, ***Hashtag***, ***Tweets*** and ***User*** data.
- **Relationships**: mainly includes the ***Has_Hashtag***, ***Mentions***, ***Posted***, ***Retweeted***, ***Reply_To*** data.
- **Everything**: includes all the data, provided for your convenience.

WARNING: **Additional data has been provided in the repositories for your convenience. Not all the data are necessarily used to perform the tasks described below.**

The detailed description of the files content is presented in the **Graph_DB_description.pdf** file in the **database** sub-repository.

Note that the data will have to be split into two sets: one for training, one for testing in the case of the application of machine learning-based methods for achieving the tasks described below. More information regarding this specific subject will be provided during the course as well as further in this document.

4 Project Organisation and Instructions

The project is to be done **by groups of 3-5 members**.

WARNING: All questions that concern the project are preferably asked on the Moodle forum that will be opened for this purpose so that the answers can be seen by all other students. Otherwise, I can answer to questions sent by mail to lynda.lechani@irit.fr. Question-answering meetings dedicated to the project will be scheduled (See Moodle).

The two following sections describe the content of what you are expected to hand-in.

4.1 Part 1: Implement a dashboard of the online crisis

Your work is expected to achieve the following objectives:

1) First of all, provide a dashboard for each event type (e.g., *flood, fire*, etc.) based on your analysis of the content and the structure of the related social sub-graphs. The dashboard should deliver (**mine from the web and then visualize**) key information about:
   - The (temporal) distribution of words and tweets across the event timeline.
   - Topics addressed in the tweets in the form of: i) word clusters built upon the k-top words based on their inverse document frequency in the corpus; ii) word embeddings; iii) tweet embeddings.
   - [*Open question*] Suggest mining any other information that would be useful in the dashboard.

2) Build a core search system (either IR or QA system) on the tweet stream across the events: given a query/question expressed using a list of keywords **Q (q_1, q_2,..q_n)**, the system should be able to return the top-k tweets ranked based on tweet relevance.

   You are expected to build a toy dataset of queries to test the system you implement.

4.2 Part 2: Predict the priority of a tweet post to feed it at the right time to the right emergency officier

Given **a tweet** post, can we predict its **priority** among the 3 category labels "*High*", "*Medium and* "*Low*"? To perform the prediction, you should combine the social web graph content, usage and structure. Schematically, any supervised learning problem can be addressed by considering the following methodology:

- Split data into training and testing subsets;
- Definition and implementation of features;
- Learning a predictive model;
- Evaluation of the model;
- Model improvement.

You will therefore adopt this methodology to carry out this multi-label classification problem. More specifically, here are some details on the work to be done.

1) **Split data into training and testing subsets.** Dividing the dataset into training and test data sets is your responsibility. You might keep a reasonable ratio between train and test (around 70% vs. 30%). You might meet the issue of imbalanced data classification and thus, you apply a suited technique for under sampling the majority class or oversampling the minority class.
2) **Definition and implementation of features.** There is no exhaustive list of features that are useful for a good prediction. This is a major step in ML projects and combines rigor and creativity. To get you started, here are some examples of features that may be useful in this context:
   - *Content-based features:* the embedding of the tweet, the embedding of the user who posted the tweet[1], characteristics of similar tweets, ….
   - *Usage-based features:* presence of opiniated words, emoticons, …
   - *Structure-based features*: popularity of tweet hashtag, popularity of the user who posted the tweet,…
3) **Learning a predictive model.** This will involve using a supervised algorithm to learn how to predict the value of the target variable (the **priority**).
4) **Model evaluation.** The metric we use to evaluate the performance of your models is the F1 metric and optionally the Cohen's Kappa metric given the imbalanced classes. Evaluate and compare these measures of at least 3 different algorithms. Another way to assess the impact of new features is to calculate the average metric score on the learning dataset via cross-validation.
5) **Model improvement.** Several refinements can be considered to improve its performance:
   - Creation or removal of features: this is often the preferred path to significantly increase the quality of a model.
   - Optimization of the hyper-parameters of the learning algorithm (optional step): once all the feature creation tracks have been considered (or when the end of the project is approaching), this technique can be useful. Indeed, all the learning algorithms can be parameterized. The values of these parameters can impact the performance of the model

[1] You can use either embeddings of the user's tweet (content) or (optional) embedding of the user node (structure) (Cf. 6.2).

and are dependent on the manipulated data. It can then be interesting to look for the best values of these parameters to boost the performance of your model. This is called the hyper-parameter optimization step. Several strategies exist (and are implemented in Python). A complete list of these strategies can be found here: https://en.wikipedia.org/wiki/Hyperparameter_optimization.

- **[Optional]** You can also explore deep neural classification models.

You will have to detail in a separate report all the instructions you used (python code), formal ML methods, results.

5 Documents and Files to hand in on Moodle

The date to hand in the all the project files will be on the Moodle Drop Zone.

All files handed in must contain the student number and name of each participant. The files have to be zipped together and are handed in as a unique zip file on Moodle: 1) scripts with comments; 2) reports in PDF format including all the answers and comments with respect to the above instructions. It is recommended to use a convenient format: provide an outline or table of contents, illustrate your results with figures and tables with captions and references, highlight the main results using brief and salient take away messages. Annexes are optional, but can be included for clarity and details if any.

6 Additional resources

6.1 Using Python nltk to analyse tweet posts

See relevant functions here:

https://pythonspot.com/tokenizing-words-and-sentences-with-nltk/

https://www.nltk.org/book/ch01.html

6.2 Graph and node embeddings

Build basic user embeddings using Nod2vec model. See relevant functions here:

https://www.kaggle.com/code/shakshisharma/graph-embeddings-deepwalk-and-node2vec/notebook

https://www.kaggle.com/code/gurkandyilmaz/node-embedding-via-node2vec

6.3 Data dictionary

The detailed description of the files content is presented in the **Graph_DB_descriprion.pdf** file in the **database** sub-repository.

Additional documentation may be posted on MOODLE if required. Feel free to ask if a specific documentation is not easy to find.