Motivate the problem well

**Introduction**

Disasters are unforeseen, calamitous event that have caused great damage and losses in term of goods and lives since the dawn of our existence. Societies have always attempted to limit the impact of such calamities by developing measures to address the initial impact and handle their consequences.

Our project focuses on this last purpose. We believe that by collecting and analyzing information about a certain disaster event, it would be easier to better understand the needs of the affected community and consequently plan the next actions to be taken in order to increase safety and limit additional damages.

Thanks to the advent of information technology, social networks can be a relevant resource to find this kind of information, as they have become the first source of *knowledge* when dealing with not ordinary events.

However, the amount of data they provide is huge and contains a lot of irrelevant messages. Follows that retrieving useful data from *them may result in* finding a needle in haystack.

Our approach proposes to *overcome these shortcomings* through Information Retrieval techniques, generating a set of words recurrent in Actionable tweets, expanding them and scoring them according to some relevance criteria that take into account also word frequency (tf-idf) in relevant and irrelevant tweets.

Information re

* Also define actionability?

**RELATED WORK**

To better understand the concept of “actionable tweets” we refer to the work done by (Derczynki et al. 2017 “Helping Crisis Responders Find the Informative Needle in the Tweet Haystack”) defining it as an “Informative message that would be helpful to the authorities or the general public with saving lives or assisting the other response teams in dealing better with the incident”.

Our research to detect actionable information from disaster data was also influenced by some information given by (Munro 2011, “Subword and spatiotemporal models for identifying actionable information in Haitian Kreyol”) and (Spasojevic et Rao 2015, “Identifying Actionable Messages on Social Media”).

**DATASET**

Our analysis is focused on Twitter. The dataset, (gently provided by the professor Cornelia Caragea), consists of a set of tweets regarding hurricanes and earthquakes that affected US and Mexico in 2017 . They have been selected by using a relevance classifier (general classifier trained on CrisisLex26 data) to identify those tweets that are relevant to a disaster and only english tweets have been kept.

It includes over 2M tweets about the Irma hurricane, 200K about Harvey, 500K about Maria, and 200K about Mexico earthquake. There are also other files with some miscellaneous data from 2 or three of these catastrophes.

Notice that the file is not labelled, hence a manual research of about 600 tweets have been made to detect some recurrences or other relevant information in actionable tweets

**METHOD**

Our experiment can be divided in 4 main phases: preprocessing, synonyms expansion, scoring and similarity evaluation.

1. Preprocessing

In this first phase, our purpose is to create a vocabulary data structure suitable for the computation of tf-idf and able to match the scored words that will be found in step 2.

To do that we decided to avoid stemming, as it could cause some loss of information and decrease the precision of our research. In fact, in our case, some words are more relevant when conjugated than in their “root form”(e.g. The word “Stuck” will retrieve more actionable tweets than its infinitive “stick”).

For the same reason, we also decided to not remove stopwords, as for example some possessive like “my” could result really informative.

So, after removing punctuation, we have collected all the words in the corpus with their frequency in a vocabulary and then scored each term using tf-idf.

1. Synonyms expansion

As the dataset was not labelled for actionable tweets, we have manually analyzed about 600 random tweets detecting the actionable ones. Doing that, we have discovered some words that are recurrent in actionable tweets and collected them into a file.

Our approach aims to filter our dataset using these words to retrieve a ranking with significant actionable tweets.

We have also decided to expand the words found with some of their synonyms (scoring them) to evaluate whether adding some other relevant word would increase accuracy.

For this purpose, we used the WordNet Library, a package within the NTLK library.

Our algorithm consists in:

* Giving to each term in the manually created list a score equal to 1;
* Expanding each term in the former list with its synonyms for each semantic field
* weighting the synonims with a score that corresponds to the maximum (if the synonym appears multiple times with different semantic meanings) “wup\_similarity” between our term and the synonym itself
* E.g ????????’

At the end the procedure a map containing all the terms in the list associated with their computed weights.

1. Scoring the results

Once we have obtained our weighted list of relevant word, for each term in that list we have performed a multiplication (COSINE SIMILARITY?) between its weight retrieved in the phase two and the tf-idf weight computed in step 1 (=tf-idf computation considering the list as “query”). The results of this calculation represent the final score on which we have based the final ranking.

After that, for every tweet that contains one or more words from our expanded list, we performed the sum of all the weights and ranked them in decreasing order according to that value.

1. Evaluation

Given that we don’t have a set of labeled data to compare with, we have performed our evaluation manually, considering the first TOP-N ranked tweets retrieved from our list and comparing the results with the ones obtained after thesaurus expansion in phase 2.

Results …………….

**CONCLUSION**

* Better without expansion
* Method quite portable and scalable
* Accuracy not bad
* Dataset not really good(some repetition and not only in english)

- LACK OF LABELED DATA

- LACK OF TIME

- PURPOSE: USE OTHER ML ALGORITHMS TO GET THE MEANING OF A TERM CONSIDERING ITS SURROUNDINGS IN THE PHRASE

- PURPOSE: USE NEURAL NETWORK

- PURPOSE: USE CLUSTER TO OBTAIN BETTER RESULT MORE RAPIDLY

1. Preprocessing: preprocess the corpus without stemming (could decrease precision) and without removing stopwords. We then score each of the terms using tf-idf.
2. Synonyms expansion:

* Retrieve the list of our manually selected words, and expand it using WordNet, a package within the NLTK library. In-depth, we expand each term with synonyms for each semantic field. The expanded list is weighted with a score that corresponds to the 'wup\_similarity' between the starting term and the retrieved synonym.
* If a synonym appears multiple times because it is a synonym to the term in different semantic fields, we average the values in order to have a single score to be assigned. (Our doubt here is whether it would be better to keep the maximum among the similarity scores, rather than the average).
* E.g.  'Lost' has 22 expansions, 7 of which correspond to the term 'lose', and 5 of which correspond to the term 'lost' itself, one for each semantic field.
* We compute the score as the sum of the similarity value between each of the 5 semantic fields of the word 'lost' and the other 22-5=17 synonyms. Since 'lose' appears 7 times as a synonym, we compute the average: sum the obtained values of similarity dividing by 7, occurrences in different semantic fields of the word 'lose'. Do you suggest we take the maximum instead of the average.

3. Scoring the results

* Once we compute these scores, should we also compute the tf-idf scores for the list? We were thinking about considering the list as a 'query', therefore assigning a tf-idf score to the query would be useful because it accounts for the document frequency of the term. Is this a suitable approach? Once we compute the tf-idf for the list we could multiply these values with the values of term similarity computed in the last step (using WordNet), this would take into account both semantic similarity and relevance of the term in the corpus.

4. Similarity

* Finally, we would rank the tweets using the cosine similarity calculated from the tf-idf scores for the corpus and the 'custom' score for the synonyms.

contains all of the tweets, roughly classified as belonging to a particular disaster based on the keywords/hashtags that they contain

but here we used a relevance classifier to identify those tweets that are relevant to a disaster (we used a general classifier trained on CrisisLex26 data) and kept only english tweets.

nowadays everyone has easy access on this information.

Related work

Dataset

Method

Results

Conclusion

Background and motivation

Nowadays social networks are one of the first sources of information when dealing with crises as users rely on them even in the most critical phases of a calamity. We believe that this is one of the best ways to help people using information retrieval. Furthermore, the topic has clearly potential that is yet to be explored

* Try to apply known algorithms and techniques to ensure people safety in case of disaster events
* Try to channel the flow of important information in order to speed up the authorities’ involvement

Dataset

* Analysis is focused on Twitter
* The dataset will be generated with focus on emotions like fear, anxiety, apprehension, sadness.

Approach

* Build the dataset with fine grained emotion detection
* Analyze it to retrieve tweets that could bring to possible actions, such as medical or psychological support.

Flag to expand/no expand query -> better no expansion