LONTAR KOMPUTER VOL. 9, NO. 2, AUGUST 2018 p-ISSN 2088-1541
DOI : 10.24843/LKJITI.2018.v09.i02.p01 e-ISSN 2541-5832
Accredited B by RISTEKDIKTI Decree No. 51/E/KPT/2017

Text Based Approach For Similar Traffic Incident Detection from Twitter

Myrna Ermawati[1], Joko Lianto Buliali[2]

[1,2] Department of Informatics, Institut Teknologi Sepuluh Nopember (ITS),
Surabaya, Indonesia
[1]myrna.winarso@gmail.com
[2]joko@cs.its .ac.id

Abstract

Microblog has been used as an information source to detect real-world event. Several related studies retrieved road traffic event based on textual content. Not only detect traffic incident, we found that it is necessary to recognize statuses with similar traffic incident content. Better representation of traffic information will help the handling of traffic incident by related parties. This study proposes text-based approach for identification of similar traffic incident from twitter posts. The proposed approach performs traffic incident information extraction and calculates information's weight based on textual similarity upon traffic incident information gained. We evaluate the proposed method by using a traffic incident information retrieval system. We used Indonesian language corpus contains traffic incident tweets data. Best average f-measure 70% was achieved by retrieval system that tested using Jaccard coefficient. Therefore text matching such as Jaccard coefficient is more suitable to be implemented in very short text document such as extracted tweet document. The experiment result gives the conclusion that the proposed approach can be implemented for identification of similar traffic incident information from Twitter.

Keywords: *text similarity, information retrieval, information extraction, similar event detection, information weighting.*

1. Introduction

Microblog has become one of the most accessible sources of information. Microblogging is part of social media that allows its users to write and share short messages (280 characters on Twitter) containing opinions, information, questions and also discussions. Microblogging services (such as Jaiku, Plurk and Twitter) are increasingly popular because of the ease of accessing and using them with the availability of social networking site apps for smartphones and tablets [1].

Microblog has also been widely used as a source of information for detection or recognition of real-world events, such as traffic incidents, earthquakes, tornadoes, wildfires, and music concerts [2]. Events can be defined as real word events occurring within a certain time period and timeframe [1][3]. In relation to traffic events or traffic information, people are also used to sharing information that occurs around them by posting a status on social media when passing on the road. Real-time traffic information such as that obtained from social networks helps users avoid traffic congestion, better plan the routes, and save fuel costs [4].

There have been many research and real-time event detection systems that utilize social media status as a source of information. Social media status and other text documents such as blogs, news sites, and emails are natural language text. Therefore we need NLP (Natural Language Processing) technique to extract meaningful information from a collection of natural language text such as Twitter post [5]. Many research related to extracting traffic information from Twitter have been conducted before, in example study by Wanichayapong et al [4], Endarnoto et al [7] and Indra [14]. Wanichayapong et al. extracted traffic information from twitter using NLP technique and syntactic analysis. Traffic information extracted was then further classified into two categories: points and links [4]. Another study by Khodra et al. extracted traffic information from Twitter and then used the extracted results as heuristic data in finding the optimal route [6]. These studies retrieve real-world event's information based on textual content.

For better information representation, it is necessary to recognize the social media status that similar, or have the same traffic incident content, with certain traffic incident information. The representation of efficient, structured and more detailed traffic incident information is expected to help the handling of events by related parties or for further data analysis. This is also to avoid repetition and storage of information with the same incident content.

This study proposes a text-based approach for identification of similar incident automatically from Twitter. We combine information extraction technique and text similarity weighting method as a hybrid, or compound, a technique to detect similar incident from Twitter post. This hybrid method assigns weight based on text similarity between traffic incident information. Our research will use this method in a retrieval system that tracks similar traffic incident information from Twitter post. The system will track previous tweets that have similarity with query tweet based on the text similarity among information entities.

We evaluate our proposed method by using Indonesian language corpus contains traffic incident tweet text. Tweet text data streams are taken from local Twitter account that reporting traffic condition in Surabaya and surrounding area. The rest of this paper is organized as follows: Section 2 presents the related study and research method including our proposed approach, design and implementation. Section 3 reports our experimental results and analysis. Finally, Section 4 concludes the paper.

2. Research Method

2.1. Literature Review

2.1.1. Information Extraction

To process and analyze text using a machine or computer, we need structured information. Information extraction as a part of NLP is a process of finding information from a collection of natural language text and producing structured information in a specific format [5][7]. Information extraction is a technique of identifying and understanding relevant sections in a text. This relevant part is called an entity [8].

The information extraction process generally finds or recognizes entities and stores into structured information in a format that suits the requirement of the application [8][15]. Information extraction is used for example in the application or question-answering system, summarization, topic extraction, the introduction of bio-medical entities such as protein names, drug product identification in medical documents, and detection of real-world events or activities [9].

The main stage in information extraction is Named Entity Recognition (NER) [15]. NER is a process that aims to find and classify the names of entities in text into named groups or attributes of structured information [4][6]. Examples of naming an entity, or an attribute of information, are 'People', 'Date', 'Organization', 'Location', 'Point', 'Department', 'Product''. Some studies classified techniques in the information extraction into 5 approaches: 1) Regression-based approaches, 2) Word dictionary approaches, 3) Rule-based approaches, 4) Machine learning-based approach, and 5) Statistical approach.

Endarnoto et al extracts traffic information from Twitter and provides visualization in mobile applications [7]. Information retrieval in this system is done by identifying entity name using rule based approach. Wanichayapong et al using the same method, the rule-based approach, but the difference is the use of a word dictionary [4]. They use word dictionaries in the tokenization process and filter tokens into several attributes, among which are verbs, points, and links. The dictionary is also used in the selection phase of twitter candidates.

2.1.2. Text Similarity

Cosine Similarity.

Cosine similarity is a method to measure the similarity of text by using the cosine value of the angle between two vectors [10][11]. The results of this calculation give a similarity value in a range of 0 to 1. Let $w(t_i, d_j)$ be a weight of term t_i in document d_j , cosine similarity value of query document q and document d_j is:

$$\cos(q, d_j) = \frac{\sum_{t_i}[w(t_i,q)] \cdot [w(t_i,d_j)]}{\sqrt{\sum|w(q\quad)|^2} \cdot \sqrt{\sum|w(d_j)|^2}} \tag{1}$$

There are many term weighting methods in the field of information retrieval and text categorization. TF-IDF (term’s frequency-inverse document frequency), or TF x IDF, is one of the popular methods used for term weighting in information retrieval. TF-IDF use weights that combine IDF factors with term frequencies TF [11]. Let $\boldsymbol{w_{i,j}}$ be the term weight associated with the term $\boldsymbol{k_i}$ and the document $\boldsymbol{d_j}$. We define $\boldsymbol{w_{i,j}}$ as

$$\boldsymbol{w_{i,j}} = \begin{cases} (1 + \log f_{i,j}) \times \log \frac{N}{n_i} & if\ f_{i,j} > 0 \\ 0 & otherwise \end{cases} , \tag{2}$$

where $f_{i,j}$ is term frequency, N is number of documents in collection, n_i is document frequency having term $\boldsymbol{k_i}$.

Jaccard Coefficient.

Similar documents are those that have the highest similarity values with the query. One of the simple techniques in calculating text similarity is to calculate the Jaccard coefficients. This coefficient is easy because we look for the same term divided by the total item of both. Jaccard coefficient is also known as a text matching method.

Using an example of the query: "ides of march" with two documents doc1: "caesar died in march", doc2: "the long march". The set q∩doc1 = {march}, q∪doc1 = {ides, of, march, caesar, died, in}. The jaccard coefficients between queries with doc1 and doc2 are shown in equations 2 and 3.

$$jaccard(q, doc1) = \frac{1}{6} \tag{3}$$

$$jaccard(q, doc2) = \frac{1}{5} \tag{4}$$

2.2. Research Question

With problems background discussed in the previous section, we may conclude two research questions:

RQ1: How to extract traffic information from twitter posts into information entities in order to detect traffic incident information.

RQ2: How to assign text similarity weight on information and use this weight to rank similar event based on textual content relevance.

2.3. Proposed Approach : Event Information Retrieval System Model

We will implement our proposed approach in the event information retrieval system. The system begins by filtering candidate tweets as described in figure 1. Candidate tweet is a tweet with traffic information content. The next stage will perform traffic information extraction. This process extracts traffic information from candidate tweet content and produces information entities. In the next process the system will search previous tweets to detect the same, or similar, event information.

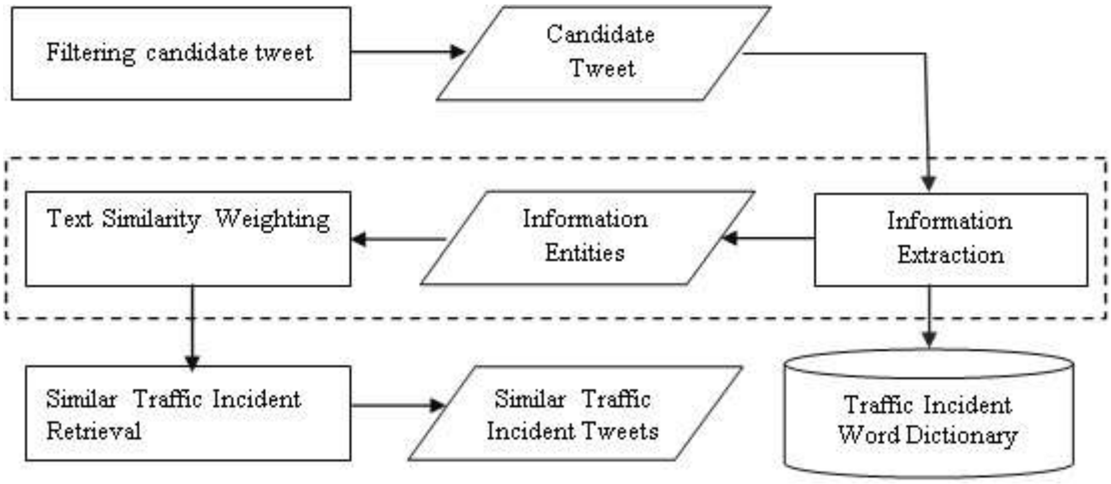

Filtering candidate tweet
Candidate Tweet
Text Similarity Weighting
Information Entities
Information Extraction
Similar Traffic Incident Retrieval
Similar Traffic Incident Tweets
Traffic Incident Word Dictionary

Figure 1. System Flow Diagram

2.3.1. Filtering Candidate Tweet

The filtering stage, as the first stage in our system, aims to recognize a raw tweet that has traffic information content, which is then called candidate tweet. A tweet becomes candidate tweet when its text, or content, consists of one of the keywords listed in pre-registered traffic keyword list. Tweets with content other than traffic information are ignored.

Our system uses 30 keywords in the candidate tweet filtering process. These keywords are obtained by observing the traffic information content tweets. A number of important words that often appear in a traffic information tweets corpus are then selected as keywords. Table 1 shows some of the keywords used in our filtering stage.

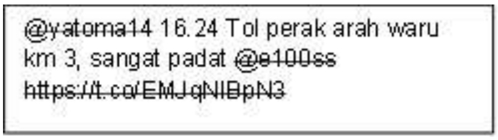
Table 1. Examples of traffic incident keyword

No.	Keywords
1	Kecelakaan
2	Tabrak
3	Jatuh
4	Mogok
5	Macet
6	Merambat
7	Tol

2.3.2. Information Extraction

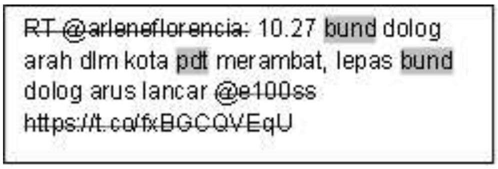
Preprocessing.

Early phase in our information extraction stage is preprocessing consists of normalization, altering word abbreviations, and case folding. Normalization removes substrings that usually appear on tweets but are not needed in our system, as mentioned and links. Figure 2 shows the example of removing mention and link, while figure 3 shows the example of abbreviation found that will be altered into its complete word. We can see some examples of preprocessing result in table 2. The second column is candidate tweet as the raw tweet to be processed. The last column shows a textual content of tweet after the preprocessing phase.



~~@yatoma14~~ 16.24 Tol perak arah waru km 3, sangat padat ~~@e100ss~~ ~~https://t.co/EMJqNIBpN3~~

Figure 2. Example of mentioned and link removal



~~RT @arleneflorencia:~~ 10.27 bund dolog arah dlm kota pdt merambat, lepas bund dolog arus lancar ~~@e100ss~~ ~~https://t.co/fxBGCQVEqU~~

Figure 3. Example of abbreviation found and will be altered

Table 2. Examples of preprocessing stage result

No.	Raw Candidate Tweet	Text after Preprocessing
1	RT @Firman_andika88: Kawasan Prempatan greges macet total tdk ada petugas mengatur lalin @e100ss	kawasan prempatan greges macet total tidak ada petugas mengatur lalu lintas
2	RT @KimNugraha004: Banjir di jalan raya pakal, sekitar 10-30 cm...padat merayap...@e100ss	banjir di jalan raya pakal sekitar 10-30 cm padat merayap
3	11.59: Info awal #kecelakaan di Exit Tol Gunungsari arah Kedurus. Ada Truk Trailer menabrak Motor. Lokasinya... https://t.co/WDy9mpHb0e	info awal di exit tol gunungsari arah kedurus ada truk trailer menabrak motor lokasinya

Dictionary Based NER.

Information extraction technique used in our experiment is a dictionary based NER [4] that utilize words dictionary. The information extraction on our system utilizes NER utility on LingPipe java. LingPipe is a toolkit in java programming for text processing by using linguistic computation. Table 3 shows examples of listed phrase and category in our dictionary.

Table 3. Examples of listed phrase and category

No.	Phrase	Category
1	pertigaan	location
2	tol	location
3	gate	location
4	tabrak	condition
5	macet total	condition
6	sepeda motor	object
7	container	object

Information Filling.

As the result of information extraction, recognized entities are then used to fill groups of information entities. This process stores extraction result into a more structured form [12][13]. Event information generally comprises entities: type of event, location, event time or period, the cause or condition, and who is involved or experiencing an event [4]. The determination of these information entities is also based on the information needs in our research. Because of these two backgrounds, this study uses 4 entities of traffic information: (1) hashtag, (2) location, (3) incident condition, (4) object. Table 4 shows an example of an information extraction result using Lingpipe's approximate dictionary. This table shows examples of extracted phrase and its category for each preprocessed text on the left column.

Table 4. Examples of extracted phrase and its category

Preprocessed Text	Extracted Phrase (information entities)			
	Hastag	Condition	Location	Object
kawasan prempatan greges macet total tidak ada petugas mengatur lalu lintas		macet total tidak ada petugas lalu lintas	kawasan prempatan greges	
banjir di jalan raya pakal sekitar 10-30 cm padat merayap		banjir padat merayap	di jalan raya pakal cm	
info awal di exit tol gunungsari arah kedurus ada truk trailer menabrak motor lokasinya	kecelakaan	ada truk trailer menabrak motor	di exit tol gunungsari arah kedurus	truk trailer motor

3. Result and Discussion

3.1. Traffic Information Tweet Data

We used data collection contains traffic incident tweets in Surabaya city and surrounding area. We evaluate our proposed approach using event information retrieval system. Therefore the corpus used in our retrieval system is Indonesian language corpus. Raw tweet data streams have taken from twitter timeline account Suara Surabaya (@e100ss). Twitter data streams have retrieved without a capture permission or data usage permission. Data crawling was done using twitter class library for java Twitter4j library.

We collected 6100 raw tweet data having a timestamp between '2017-11-17 15:49:52' and '2017-12-25 10:04:37'. From the filtering stage, we obtained 2360 candidate tweets containing traffic incident information. Therefore after information extraction stage, we had 2360 traffic tweets, saved with its information entities such as showed in table 4, as a corpus, or document collection, for similar traffic incident detection in our traffic information retrieval system. We also manually observed, collected and labeled several candidate tweets used as query tweets and its relevant tweet for. Next subsection will give more brief explanation about the evaluation including query tweet tested and the evaluation result.

3.2. Experiment

Our experiment performed top-1 retrieval system comparing weighting method using three different methods for text similarity measurement: 1) cosine similarity using TF (term's frequency) term weighting, 2) cosine similarity using TF-IDF (term's frequency-inverse document frequency) term weighting, and 3) jaccard coefficient. Our idea is to analyze which text similarity measurement is more suitable for a very short text such as traffic entities extracted from a tweet which already short in a text.

This test has been done using 20 query tweets which have only one relevant previous tweet. Query tweet is a selected tweet that has content of traffic incident and has another related tweet named relevant tweet. A relevant tweet is a related tweet contains similar traffic incident information content with the query tweet. We had manually observed, collected and labeled several query tweets and its relevant tweet. We collected a small number of query tweets due to the limited number of real traffic incident information posted that have a relevant tweet in our tweets data collection.

While testing a query in the retrieval system, tweet documents in the corpus are ranked in decreasing order of their degree of similarity. We calculated average precision, recall, f-measure, and average count of relevant tweet achieved the top-1 position as retrieval output. Table 5 shows three examples of query tweet and its single relevant tweet.

Table 5. Examples of query tweet and it's relevant tweet

Id.	Query Tweet	Relevant Tweet
Q2	RT @kang_de2n: @e100ss ada kecelakaan tunggal di tol legundi arah mojokerto KM 716.200. Truk muat kayu pecah ban muatan tumpah ke badan ja…	11.31: Info awal #kecelakaan di Tol Krian - Mojokerto KM 716.800. Truk muat kayu pecah ban, kemudian terguling di... https://t.co/v1KVxD0eac
Q3	Macet total Krian Surabaya 2 arah, truk as patah di Sidorejo. Cari alternatif. (rs) https://t.co/V5azxGvTNA	RT @xenopchilla: @e100ss Ini lho penyebab macet dua arah di raya sidorejo... https://t.co/Nd4ISgpuYH
Q11	RT @andhikanoviandy: @e100ss waspada , ada truk pecah ban sebelum res area tol waru arah sidoarjo	RT @josuryana: @e100ss ada truk berhenti krn ban pecah di tol km 12 waru arah sidoarjo

By using query tweet having only one relevant tweet, the experiment evaluated the retrieval result based on first rank output. Table 6 and 7 show experiment result using proposed method tested using a retrieval system. Table 6 shows the real rank position of relevant tweet returned of query ID Q2, Q3, and Q11. As mentioned above, CS-TF is cosine similarity using TF term weighting and CS-TF.IDF is cosine similarity using TF.IDF term weighting. Rank number zero means retrieval output rank was out of top-20 output list. Low rank of relevant tweet returned when testing query ID Q3 due to its relevant tweet less informative. As we can read relevant tweet ID Q3 in table 5, there is no information about the cause of traffic jam at raya sidorejo because it is indicated by the picture in its hyperlink and not in its text such as "truk as patah".

Table 6. Retrieval output: relevant tweet rank of query ID Q2, Q3, Q11

Query ID	Relevant tweet rank number		
	CS[1]-TF[2]	CS[1]-TF[2].IDF[3]	Jaccard Coef.
Q2	1	6	1
Q3	15	0	5
Q11	1	1	1

[1]cosine similarity [2]term's frequency [3]Inverse document frequency

Table 7. Average performance value of 20 query tweets

Retrieval Performance	Performance Value (%)		
	CS[1]-TF[2]	CS[1]-TF[2].IDF[3]	Jaccard Coef.
1st Rank Total Count	12	3	**14**
1st Rank Ratio	0.6	0.15	**0.7**
Average F-Measure	60%	15%	**70%**

[1]cosine similarity [2]term's frequency [3]Inverse document frequency

Table 7 shows the performance values of retrieval output based on top-1 retrieval using 20 query tweets. Total count of first-rank achieved higher value when we use jaccard coefficient. Best average f-measure 70% was achieved by retrieval system that tested using jaccard coefficient. The experiment showed that our retrieval performance achieved a good result in retrieving similar traffic incident tweet.

IDF term weighting comes from idea regarding the term specificity. The more a term occurs in many documents, the term becomes less specific depending on its meaning. This statistical term specificity is the inverse of the number of documents in which the term occurs. While TF and Jaccard coefficients are computed on a per document basis, term weighting IDF is computed over all the collection. This is the reason why TF-IDF term weighting achieved low retrieval performance compared to TF and Jaccard coefficient in our experiment. A document in our retrieval system is a quite short length, combined phrases extracted from a twitter post that already short in a text. Then we only need a similarity measurement, such as Jaccard coefficient, that simply looks the same term between these two short documents. Table 7 shows that text similarity measurement on a short text using jaccard coefficient has a better result than cosine similarity with TF and TF.IDF. Therefore jaccard coefficient is more suitable to be used as text similarity measurement for identification of similar traffic incident information from twitter.

4. Conclusion and Future Works

We had studied and analyzed our text based approach to track similar traffic incident information. The experiment showed that retrieval performance results achieved a good result in retrieving similar traffic incident tweet. Based on the retrieval performance result we make a conclusion that our text based approach can be implemented for identification of similar traffic incident information from twitter. The experiment result also gives conclusion that text matching such as jaccard coefficient is more suitable to be implemented in very short text document such as extracted tweet document.

Text similarity in our study has not considered the existence of different words with the same meaning in a traffic incident, for example the term ‘tabrakan beruntun' and 'kecelakaan beruntun'. Therefore the next research may overcome this problem with semantic analysis.

References

[1] F. Atefeh and W. Khreich, “A survey of techniques for event detection in Twitter”, Comput. Intell., vol. 31, no. 1, pp. 132–164, 2015.

[2] T. Sakaki, M. Okazaki, and Y.Matsuo, “Tweet analysis for real-time event detection and earthquake reporting system development”, IEEE Trans.Knowl. Data Eng., vol. 25, no. 4, pp. 919–931, Apr. 2013.

[3] J. Allan, “Topic Detection and Tracking: Event-Based Information Organization”, Norwell, MA, USA: Kluwer, 2002.

[4] N. Wanichayapong, W. Pruthipunyaskul, W. Pattara-Atikom, and P. Chaovalit, “Social-based traffic information extraction and classification”, in Proc. 11th Int. Conf. ITST, St. Petersburg, Russia, pp. 107–112, 2011.

[5] E. D'Andrea P. Ducange B. Lazzerini F. Marcelloni "Real-time detection of traffic from twitter stream analysis" IEEE Trans. Intell. Transp. Syst. vol. 16 no. 4 pp. 1-15, Aug. 2015.

[6] Khodra, M.L., Purwarianti, A., “Optimal Path Finding based on Traffic Information Extraction from Twitter”, Prosiding International Conference on ICT for Smart Society 2013, Jakarta, 2013.

[7] Endarnoto, S., Pradipta, S., A.S, N., & Purnama, J, “Traffic Condition Information Extraction & Visualizations from Social Media Twitter for Android Mobile Application”, ICEEI (pp. 1-4). IEEE, 2011.

[8] Jiang, J., “Information Extraction from Text, in Mining Text Data”, Springer, 2012.

[9] A. Hotho, A. Nürnberger, and G. Paaß, “A brief survey of text mining”, LDV Forum-GLDV J. Comput. Linguistics Lang. Technol., vol. 20, no. 1, pp. 19–62, May 2005.

[10] C. D. Manning, P. Raghavan, and H. Schutze, “Introduction to Information Retrieval”, Camridge: Cambridge University Press, 2008.

[11] Fauzi, M. Ali; Arifin, Agus; Yuniarti, Anny, “Term Weighting Berbasis Indeks Buku dan Kelas untuk Perangkingan Dokumen Berbahasa Arab”, Lontar Komputer : Jurnal Ilmiah Teknologi Informasi, vol.5 no.2, Aug.2014.

[12] Khodra, M.L., Purwarianti, A., “Ekstraksi Informasi Transaksi Online pada Twitter”, Jurnal Cybermatika, vol.1, 2013.

[13] Khodra, M.L., Purwarianti, A., “Optimal Path Finding based on Traffic Information Extraction from Twitter”, Prosiding International Conference on ICT for Smart Society 2013, Jakarta 2013.

[14] N. Indra, “Sistem Pemberi Tahu Kemacetan Lalu Lintas di Kota Bandung Berbasis Media Sosial”, Laporan tugas akhir, InstitutTeknologi Bandung, Bandung: Program Studi Teknik Informatika.

[15] Manning, C., Information Extraction and Named Entity Recognition.California: Stanford University. 2012.