2011 International Conference on Electrical Engineering and Informatics
17-19 July 2011, Bandung, Indonesia

Traffic Condition Information Extraction & Visualization from Social Media Twitter for Android Mobile Application

Sri Krisna Endarnoto[#1], Sonny Pradipta[#2], Anto Satriyo Nugroho[*3], James Purnama[#4]

[#]*Faculty of Information Technology, Swiss German University, Tangerang, Indonesia*
EduTown BSDCity, Tangerang 15339, Indonesia
[1]sri.endarnoto@student.sgu.ac.id
[2]sonny.pradipta@student.sgu.ac.id
[4]james.purnama@sgu.ac.id

[*]*Center for Information and Communication Technology (PTIK)*
Agency for the Assessment and Application of Technology (BPPT)
Jalan M.H. Thamrin No.8, Jakarta 10340, Indonesia
[3]asnugroho@ieee.org

Abstract— **Traffic jam in Jakarta, Indonesia has become a crucial problem for the society. A Traffic Management Center has been built by the police, in this case Polda Metro Jaya to help people to get the latest information regarding traffic jam. Twitter has been used by TMC Polda Metro Jaya to spread the news of traffic. In this paper, information extraction technique is used to get the data of traffic, so that the traffic information can be presented in map view as a mobile application of Android. Early experiment with limited vocabulary and rules has showed promising result.**

Keywords— **Information extraction, Natural Language Processing, mobile application**

I. INTRODUCTION

In Jakarta, Indonesia, traffic jam is one of the biggest problems for society. A traffic management center has been built by the police. The main objective of TMC (traffic management center) is of course to help people to get the information of traffic. Social media such as Twitter has been used to spread the news of traffic. With its limitation, Twitter doesn't provide good user interface in the case of traffic condition.

The main objective of this project is to help people in Jakarta, Indonesia to get the news of traffic from a reliable source with a very nice presentation by developing a system that can extract the information of traffic from Twitter account of TMC Polda Metro Jaya (police unit in Jakarta) to be presented in a map view by using Google Map and implement it in Android-based mobile application. Twitter account of TMC Polda Metro Jaya (@TMCPoldaMetro) can be accessed via: http://www.twitter.com/#!/tmcpoldametro.

Natural Language Processing can be used to extract that information from Twitter of TMC Polda Metro Jaya. The tasks that will be handled are basically to analyze a tweet, get certain information that we need regarding the traffic, and use those information for Android mobile application which will display the traffic condition in map form, with 3 different colors for different traffic conditions. Green for normal, yellow for crowded, red for jammed.

Android is chosen because of the increasing popularity of Android-based mobile phone in Indonesia.

Our paper is related to parsing Indonesian language such as parsing Indonesian with Context Free Grammar [1] and probability parsing for Indonesian language [2]. Our paper is also related to text mining from Twitter such as key phrase extraction from Twitter [3] and characterizing users and tweets [4].

But our paper is mostly related to information extraction from Indonesian sentences such as extracting information from e-job marketplace [5].

This paper will present the process of our project and the result. In section 2, we discuss the method that we use. The experiments and the result will be presented in section 3, followed by conclusion and future work.

II. METHODS

The main idea is to extract information of traffic from TMC's Twitter. There are several critical information that the system must get and store in the database. Android-based mobile application should get those data, so the information can be presented in map view.

The information extraction system will run continuously and should not be shut down. It must keep listening to the data, checking the database if there is new tweet to be processed. So basically it works in circle, and will be slept for some amount of time if there is no update.

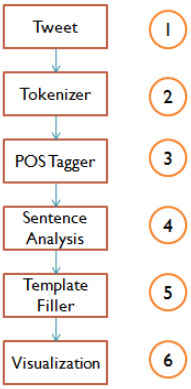

Fig. 1 Information extraction process

Several steps are taken in order to make this system works. The information extraction follows process that is shown in Fig. 1.

A. Fetch tweet

The first process in the circle is to fetch the tweet from TMC's Twitter. Update from TMC's Twitter account will be stored in a table in the database. The system will work with tweets from this table by fetching new updates. Once the tweet has been processed, the flag in that respective record in the table will be changed.

B. Tokenization

After fetching tweet, the next thing that will be done by the system is to tokenize the tweet. The system will break the tweet into words, so called tokens. These tokens will be used during the next step in the circle, which is Part of Speech tagging.

C. POS Tag

Part of Speech (POS) tagging is basically classifying /tagging each token which is produced by the previous step, based on pre-defined word class/POS name. With some rules which is defined in the database, tagging process will search whether each of the token match with one of the word in the vocabulary collection in the database for each possible word class/POS name. This phase, together with sentence analysis and template filling are steps in [6].

D. Sentence Analysis

Once each token has been tagged, those tokens will be analyzed by the system by using some pre-defined rules. Rules that will be used are basically gathered from the analysis of TMC's tweet format. Those rules will be used to gathered useful information regarding traffic, such as:

- Time
- Origin
- Destination
- Traffic condition

E. Template Filling

There are 4 main informations that we need to get from the information extraction system, so the Android-based mobile application will be able to present it in map view. Time, origin, destination and traffic condition. Those 4 informations will be stored in the database. Those attributes of information that we want to extract is basically a template, and in this step the database will be filled by the information that are extracted from the tweet.

F. Visualization

The visualization part is the task of Android-based mobile application. The application on handset will communicate with the server in which the database stored, and will ask for the condition in a certain area. Then, the application will present the traffic condition of that area based on the information from tweet that has already been extracted by the information extraction system. The presentation of traffic condition will be based on 3 status with 3 different colors.

III. Experiments

A. Information Extraction System

From the information extraction system, early experiment uses sample of 100 tweets from Twitter account of TMC Polda Metro Jaya.

Fig. 2 shows 3 example of TMC's tweet that we use in the experiment. First example means at 06:39, from Fatmawati (name of place) to Blok A (name of place) traffic crowded nearly jammed. Second example means at 06:23, from Joglo Main Road to Pos Pengumben (name of place), traffic jammed. Third example means at 10:51, from Bekasi (name of place) to Cawang (name of place), traffic crowded nearly jammed.

POS Tag phase uses Tagset shown in Table 1. And rules for information extraction are constructed manually by analyzing random tweet of TMC. Example of those rules is shown in Fig. 3.

```
06:39 Fatmawati arah ke Blok A padat merayap.
06:23 Jl.Raya Joglo arah Pos Pengumben lalin tersendat
10:51 Bekasi arah Cawang lalin padat merayap
```

Fig. 2 Sample of TMC's tweet

TABLE I
TAGSET

No	POS	POS Name	Example
1	AJ	Adjective	Ramai (crowded), Macet (jammed)
2	AT	Adjective Time	06:50
3	AV	Adverb	Sangat (highly)
4	CJ	Conjunction	Dan (and), Lalu (then)
5	N	Noun	Lalin (traffic), Arus (stream)
6	NP	Noun Place	Pondok Indah, Bintaro
7	P	Preposition	Di (at), Ke (to), Dari (from)
8	V	Verb	Merayap (crawling), Terjadi (happening)

```
AT NP P NP AJ V
AT NP P NP N AJ
AT NP P NP N AJ V
```

Fig. 3 Example of rules

```
Time : 06:39
From : Fatmawati
To : Blok A
Condition : padat merayap
```

Fig. 4 Result from the system

Fig. 3 shows 3 examples of rules that we used in our experiment. Those rules represent the sequence of appearance of POS name in a tweet. For example if we know that the sequence matches the first rule (AT NP P NP AJ V) or third rule (AT NP P NP N AJ V), then we can get AT (adjective time) as time, first NP (noun place) as origin, second NP as destination and AJ+V (adjective followed by verb) is the traffic condition. As for the second rule, we can get AT as time, first NP as origin, second NP as destination and AJ as traffic condition.

When the first experiment began, around 50% out of 100 worked well. The problem that was found is mostly Out of Vocabulary (OOV) and Out of Rules (OOR). Fig. 4 shows example of result, captured from the system showing time, from, to and condition. Condition "padat merayap" means crowded/crawling nearly jammed.

Out of Vocabulary is the condition where the information extraction system can't find the match for a token in the list of vocabulary in the database. This is actually can be handled by some rules, but consider a token of noun place that needs coordinate of that respective place in order to present it in map view, tagging it as a "noun place" without the exact match in the database might be possible, but without the coordinate, it would be useless.

Out of Rules is the condition where the information extraction system can't find the exact way of finding how to get the needed data. This is related to the form of the sentences. More form of sentences means more rules.

Another problem that was found in the first experiment is that there were some cases where the tweet shows that the traffic condition is not only from origin to destination but also in reverse style both in one tweet. Means, we need to handle those kinds of tweets. Fig. 5 shows the example of this case. The example means at 06:43, Fatmawati Road to Panglima Polim (name of place), traffic crowded nearly jammed, in reverse, crowded.

A simple solution for that problem was to add another POS name which will be used to identify if that kind of case happens. We know that words such as "sebaliknya" (in reverse), "di kedua arahnya" (both directions) and "arah sebaliknya" (in the other direction) are such an indicator for such cases. So, new POS name called "*Indicator*" is used to store such words. Once we find the indicator, the system will generate 2 reports of traffic condition.

```
06:43 Jl Fatmawati arah Panglima Polim lalin Padat Merayap, sebaliknya Ramai
```

Fig. 5 Example of unhandled form

```
Report 1

Time : 06:43
From : Jl Fatmawati
To : Panglima Polim
Condition : Padat Merayap

Report 2

Time : 06:43
From : Panglima Polim
To : Jl Fatmawati
Condition : Ramai
```

Fig. 6 Result from the system for 2 reports

Fig. 6 shows the example of result from the system for the previous unhandled tweet. It shows that after we added new POS name of "*Indicator*", the system has successfully read that tweet and reported 2 traffic conditions. Showing that in one direction, the condition is "padat merayap" (crawling, nearly jammed) and in the other direction, traffic condition is "ramai" (crowded).

The next experiment, the rate rose to around 70% out of 100. The problems found are still OOV and OOR. So the list of rejected tweet was analyzed and the dictionary was updated.

The last experiment showed that the problem with previous rejected tweets has been solved. Among 100 sample tweets, all of them can be identified and extracted.

B. Android-based Mobile Application

From the Android, the experiment was to present the previous information that we get from the information extraction system as a map view in Android-based mobile application.

The experiment used Android Virtual Devices or emulator with Google API version 2.2 and API level of 8. The Android application will communicate with the server, in which we store the database of traffic condition from the information extraction system. Fig. 7 shows the Android emulator interface.

Fig. 7 Android emulator

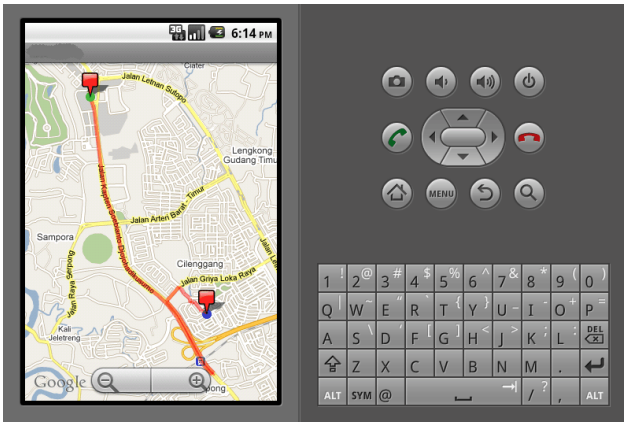

Fig. 8 Result of the traffic condition presentation

Android application communicates with the server by implementing web service. Android application in the handset will request for the actual traffic information to the server. The server will response in JSON format, including the information of place, its coordinate and the traffic condition.

Then, Android application will display map view with the information of traffic condition. Fig. 8 shows the result of the traffic condition presentation.

The experiment of Android mobile application was well worked. The mobile application has been able to display the traffic information in map view.

IV. Conclusions

To be able to present traffic condition in Jakarta, Indonesia with Twitter of TMC Polda Metro Jaya as a source, we can use information extraction method to get the main information of the traffic, and use those informations to present it in map view in the Android-based mobile application.

The result of our project has shown that the information extraction system worked well to extract traffic information from Twitter, and the Android-based mobile application worked well also to display the information in map view.

Our experiment has also shown that there is a chance the system will not read the input tweet. This is mainly because of Out of Vocabulary and Out of Rules. So the dictionary, both for vocabulary and rules need to be updated regularly, especially for the database of place and its coordinate.

Future work of our project are integrating the whole system with the profiling based mobile advertisement, new feature of the Android-based mobile application and new source of traffic information such as Twitter of National Traffic Management Center which means larger scope of information, not only the city of Jakarta but also other area of Indonesia as well.

Acknowledgment

This work is supported by Franciscus Chandra Pawitra, Rinto Priambodo and Rindang Septyan from Jatis Mobile for the help in developing Android application.

References

[1] R. Paramita N, D. H. Widyantoro, and A. Purwarianti, *INAGP : Pengurai Kalimat Bahasa Indonesia Sebagai Alat Bantu Untuk Pengembangan Aplikasi PBA,* Institut Technology Bandung Indonesia, 2009. [Online]. Available: *www.informatika.org/~**ayu**/2009parser.pdf*

[2] R. H. Gusmita and R. Manurung, "Some Initial Experiments with Indonesian Probabilistic Parsing," *in Proc. 2'nd International Malindo Workshop Cyberjaya Malaysia*, 2008.

[3] X. Zhao, J. Jiang, J. He, Y. Song, P. Achanauparp, E. Lim and X. Li. "Topical keyphrase extraction from Twitter," *in Proc. of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, 2011.

[4] D. Ramage, S. Dumais and D. Liebling. "Characterizing Microblogs with Topic Models," *in Proc. International AAAI Conference on Weblogs and Social Media*, 2010.

[5] D. H. Widyantoro and Y. Wibisono, *Information Extraction for E-Job Marketplace,* International Conference TSSA, 2007. [Online]. Available: http://fpmipa.upi.edu/staff/yudi/tssa-2007-dwi-yudi-information-extraction-for-ejob-marketplace.pdf.

[6] S. M. Weiss, N. Indurkhya, T. Zhang and F. J. Damerau. *Text Mining, Predictive Methods for Analyzing Unstructured Information.* USA : Springer, 2005.