Real-time Natural Language Processing for Crowdsourced Road Traffic Alerts

CONFERENCE PAPER · AUGUST 2015

4 AUTHORS, INCLUDING:

CD Athuraliya
WSO2
1 PUBLICATION **0** CITATIONS
SEE PROFILE

Madhawa Gunasekara
University of Sri Jayewardenepura
1 PUBLICATION **0** CITATIONS
SEE PROFILE

Srinath Perera
WSO2
41 PUBLICATIONS **193** CITATIONS
SEE PROFILE

Real-time Natural Language Processing for Crowdsourced Road Traffic Alerts

C.D. Athuraliya*†[1], M.K.H. Gunasekara*†[2], Srinath Perera*[3], Sriskandarajah Suhothayan*[4]
*WSO2 Inc., Colombo, Sri Lanka.
[1]chathurike@wso2.com
[2]madhawag@wso2.com
[3]srinath@wso2.com
[4]suho@wso2.com

***Abstract*—Out of many issues we face in transportation today, road traffic has become the most crucial issue that directly affects our lives and economy. Despite of many implemented and progressing solutions, this issue seems to be remaining in a significant level in many countries and regions. Instead of fully relying on solutions provided by the authorities, public has come up with different approaches to deal with this problem. In this study we are focusing on one such solution which effectively uses a popular social networking service, Twitter. But still this crowdsourced traffic alert service has a limitation due to its nature; the natural language representation. We are trying to cope with this limitation by introducing a real time natural language processing solution to generate machine readable road traffic alerts. We observe many potentials of transforming this raw data into a machine readable format. An architecture that can effectively capture, transform and present this data has been proposed in this study and it has been implemented in a prototype level to demonstrate the uses of such a model. We expect to see extended models that can handle similar issues in future by combining multiple fields of information technology.**

Keywords*—Traffic monitoring, natural language processing, complex event processing.

I. Introduction

Success of modern day enterprises and businesses is highly relied on how they process massive amounts of data generated everyday. Terms such as data mining, knowledge discovery and big data have become more familiar in our day-to-day life. Still it is said that we are "drowning in data yet starving for knowledge". Thousands of sources generate data which can be useful in creating knowledge, nevertheless we are still in a struggle to extract useful information out of this data. Particularly with the emergence of social media, public has gained the potential to generate massive amounts of data everyday. This data is generated in a number of formats such as text, image and video, making knowledge extraction more complicated.

At the same time road traffic has become a major issue, mainly in developing countries. Different solutions have been adopted and utilized but this issue is not seemed to be resolved fully, due to the complexity of its causing factors. Especially with growing number of vehicles, road traffic has become a severe problem in many countries. It has been identified that this problem directly affects country's economy and development due to the waste of resources such as fuel and time. Instead of depending on solutions provided by authorities, people have become aware of using technology to resolve most of the issues they are facing today. Furthermore, these solutions have been proven to be success stories in number of cases irrespective of their field.

In this study we focused on one such solution emerged with the use of social media. It uses an online social networking service called Twitter which is popular for dynamic content publishing. Users publish updates on different topics such as current affairs, news, politics and personal interests via 140 character messages called tweets. Twitter is also popular for trends, created by users mainly by using hashtags (#). Mentions (@) are used to tag other Twitter users in a tweet. The service we focused in this study uses these tools efficiently to publish alerts on road traffic to other Twitter users and followers.

II. Background

Road.lk [1] is a website that provides localized traffic alerts from a Twitter feed with a specific mention. This website has its own Twitter account with the ID road_lk [2]. If a particular person is experiencing road traffic or if he/she has information regarding road traffic, that person can tweet about that with a mention to road_lk Twitter account. The same tweet will be manually retweeted by road_lk account. Thus all the Twitter users who are following road_lk account will get that particular traffic alert nearly in real-time. We can observe that this model works effectively because of many Twitter users who post traffic updates with road_lk mention. This feed was identified as a potential source to extract information on road traffic in real-time. Furthermore the reliability of this source is maintained by the model itself due to higher number of publishers. If there is high traffic in a particular area, we get more traffic alerts from different users. Hence the potential of this source is significant to a country like Sri Lanka specially because of the unavailability of high tech traffic monitoring systems. By following these crowdsourced alerts a passenger or a driver can avoid high traffic areas and also can get notified on noteworthy incidents.

But still this model has several limitations such as connectivity requirement and unavailability of proper alert mechanism except Twitter feed or road.lk website. Another notable limitation is imposed by the way users post their traffic updates. Twitter users use natural language to post traffic updates. An alert format has not been imposed by road_lk to post tweets on road traffic. Processing tweets can be made more straightforward by imposing such a format, but it can significantly reduce the flexibility of sharing updates on road traffic. We identified the use of natural language processing as the best way to tackle this problem.

By combining NLP and CEP tools, we implemented a prototype solution which accommodates three use cases given below.

†These authors contributed equally to this work.

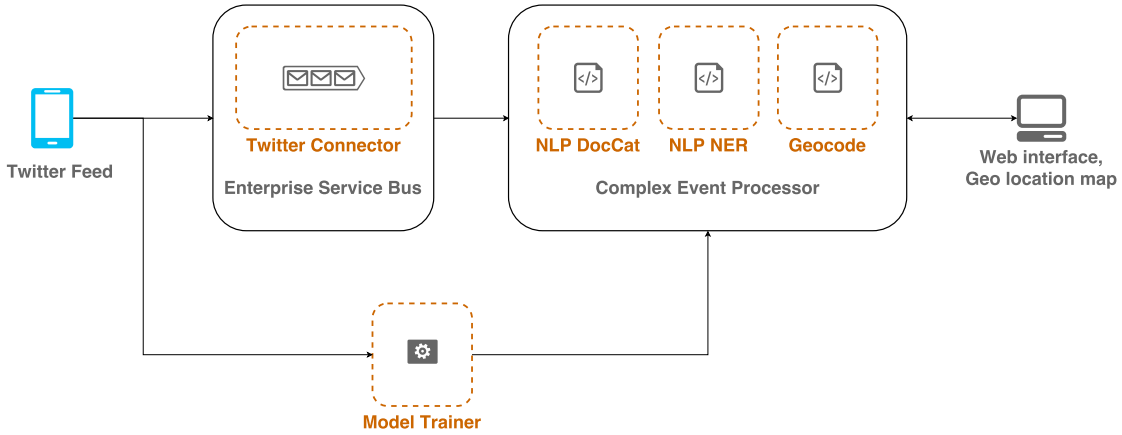

Fig. 1: Architecture of the solution

1) Real-time road traffic feed and geo location map
2) Traffic search within an area
3) Traffic alert subscription

But it is evident that this information can be processed and interpreted in different ways other than the above given use cases. Possible extensions to this study are detailed in the final section of this document.

III. Related Work

Traffic monitoring and analysing became an active research area with the development of intelligent transportation systems. Efforts have been made by researchers to provide solutions to certain issues we face in traffic monitoring systems. High quality traffic information is required for effective traffic monitoring and analysing. Several data collection methods are available [3] for real-time traffic monitoring such as traditional on-road sensors, floating car data (FCD) [4], vision systems and crowdsourced social media. FCD and crowdsourced data sources are comparatively more cost-effective methods for traffic data collection. But crowdsourced data sources contain unstructured data and they require certain data cleaning and preprocessing beforehand the actual use. Ritter et al. [5] applied unsupervised clustered modelling for Twitter corpus of 1.3 million conversations. Wang et al. [6] proposed a real-time traffic alert and warning system based on a Latent Dirichlet Allocation based approach to classify traffic related tweets by using Twitter as a crowdsourced data source. Apart from these methods, most researchers have used vision systems as their data source. For an example we can consider the traffic scene analysis system which was developed by combining a low-level machine vision-based surveillance system with a high-level symbolic reasoner based on dynamic belief networks [7].

Traffic information is considered as an important measure about transportation in most developed countries. Hence countries such as Spain [8] and Finland [9] maintain online traffic information systems to collect traffic data and share them with public. Furthermore, we can observe some existing mobile applications for traffic monitoring like Twitraffic [10]. Twitraffic is a commercial solution which can be used within UK, and it also uses Twitter feed as their data source. Users can subscribe to locations to receive traffic related tweets. But these tweets are unprocessed and they can contain misleading information.

IV. Solution

After analysing above mentioned use cases, we had to develop an architecture for a solution which addresses multiple requirements. Mainly two architectures were implemented and tested in prototype level to select the optimal approach. In both architectures, multiple tools were utilized to retrieve, process and present information in most efficient way. Overview of the implemented architecture and high level system pipelines are is given in figures 1, 2, 3 and 4 respectively.

A. Feed Retrieval

Since the solution proposed by this study is directly based on Twitter social networking service, it was required to access this social network via its API. We retrieved existing feed for dataset generation and real-time feed stream for alert generation via REST and streaming APIs respectively. Real-time data retrieval was supported by WSO2 Enterprise Service Bus [11] Twitter connector with inbound endpoints.

B. Application of Natural Language Processing

Despite the fact that road_lk Twitter feed is a reliable data source to generate real-time road traffic alerts, its extent is largely constrained by natural language representation. If we can transform this data into a machine readable representation, we can use the full potential of this source for a better solution. In this study we propose a natural language processing (NLP) model to address this problem. NLP is an area derived from fields such as machine learning and human computer interaction which is concerned in removing or reducing the language gap between humans and computers by introducing tools and techniques that enable natural language understanding to computers. This objective has been considerably achieved by cutting edge NLP tools and we can see their applications in devices such as mobile phones and tablet computers. In this study, we were interested in extracting two entities from a tweet namely, location and traffic level. Timestamp which is essential in providing useful traffic alerts was already available in the dataset. To extract these two entities, our approach was to utilize NLP tools. Ahead of extracting these two entities, a tweet was needed to be classified to identify whether it is a traffic alert or not. Users tweet on topics other than road traffic with mentions to road_lk. The NLP tasks required to classify and extract interested fields from a tweet can be listed as below.

1) Tweet categorization

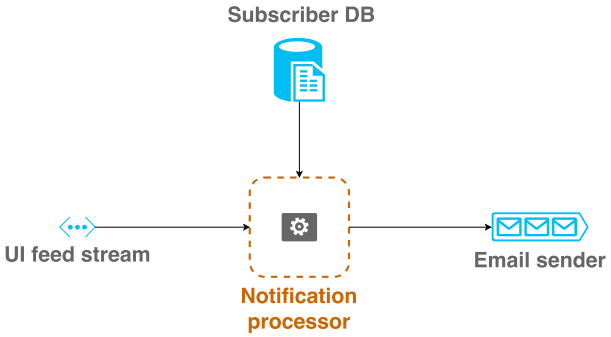

Fig. 4: Pipeline for email notifications

2) Location extraction
3) Traffic level extraction

In NLP terminology, first task is a document categorization task and the latter two are name entity recognition (NER) tasks. To implement these NLP tasks we used Apache OpenNLP toolkit [12] which is a machine learning based toolkit for the processing of natural language text. It supports common NLP tasks such as tokenization, sentence segmentation, part-of-speech tagging, named entity extraction, chunking, parsing, and coreference resolution. Out of these we only used previously mentioned two functions for our three NLP tasks. Expectation-maximization, a semi-supervised learning algorithm, [13] plays a major role at the core of these functions. Three separate models were trained for each task by using the dataset generated by retrieved Twitter feed. Initial dataset which contained around 3000 tweets was split and manually tagged to train these models. A custom tokenizer was implemented to tokenize tweet text to words. The main reason to use a custom tokenizer instead of the tokenizer available in OpenNLP was due to complex name entity tagging. For an example some street names and city names became meaningless when they were tokenized as separate words for location NER task. For traffic level NER task, a predefined set of words was selected to tag in tweet texts. In the training stage, traffic level NER model learns related words which are mostly used by users to express traffic level. Additionally we had to consider factors such as spelling mistakes, informal language and abbreviations when training traffic level NER model. These models were deployed in Siddhi language extensions, explained in next section.

C. Application of Complex Event Processing

We had to consider another important property in this particular data source when processing information; it was required to process this Twitter feed in real-time. There can be a use case of batch processing, but to generate useful alerts on road traffic, real-time processing was essential. Our approach to this requirement was complex event processing (CEP). It is a field, concerned in processing data from multiple sources in real-time. A single data input is considered as an event and a continuous data input is identified as an event stream in CEP context. We used WSO2 Complex Event Processor [14] as the CEP tool to analyse and process Twitter feed input stream. At the core of WSO2 CEP the actual event processing is done by Siddhi Query Language (SiddhiQL) [15], [16]. It is designed to process event streams and identify complex event occurrences. Siddhi queries define how to process and combine existing event streams to create new event streams. When deployed in the Siddhi runtime, SiddhiQL queries process incoming event streams as specified by the queries and generate output event streams according to the query definition. SiddhiQL was extended to address our NLP requirements using language extensions. Three extensions were implemented for following real-time processing tasks.

1) Tweet categorization
2) Name entity recognition
3) Geocoding

Geocoding extension converts the locations extracted from tweet content into geo coordinates which are required in setting marker on geo map and for calculating nearby geo area. WSO2 CEP specific structures called event flows and execution plans were deployed to process incoming Twitter feed in real-time. The functionality of the use cases such as traffic feed and traffic search were implemented within CEP instead of having a dedicated back-end server for web UI and for geo location map. Searching functionality uses a time-based Siddhi window [17] to retrieve traffic in nearby geo area within a predefined time period. A few deployed Siddhi execution plans are given below.

```
from classifiedStream#transform.nlp:getEntities(
    convertedText,4,true,"/_system/governance/en-
    location.bin")
select * insert into templocationStream;
from classifiedStream#transform.nlp:getEntities(
    convertedText,1,false,"/_system/governance/en-
    trafficlevel.bin")
select * insert into temptrafficlevelStream;
from S1=classifiedStream, S2=temptrafficlevelStream,
     S3=templocationStream
select S1.createdAt as time, S2.nameElement1 as
    trafficLevel, S3.nameElement1 as location1, S3.
    nameElement2 as location2, S3.nameElement3 as
    location3, S3.nameElement4 as location4
insert into locationsStream;
```

```
from uiFeedStream#window.time(120 min) as
    trafficFeed join SearchEventStream as request
on (trafficFeed.latitude < request.latitude + 0.018
    and trafficFeed.latitude > request.latitude -
    0.018 and trafficFeed.longitude < request.
    longitude + 0.027 and trafficFeed.longitude >
    request.longitude - 0.027)
select trafficFeed.formattedAddress, trafficFeed.
    latitude, trafficFeed.longitude, trafficFeed.
    level, trafficFeed.time
insert into searchResult;
```

V. Results and Evaluation

By implementing the approach specified in previous section, we obtained a system which can successfully retrieve and transform a Twitter feed into machine readable format. Using this outcome we implemented a web based interface to demonstrate the functionalities of our implementation. Users can interact with this interface and make use of the use cases we presented in section II. Figures 5, 6 and 7 correspond to each use case namely, traffic feed with geo location map, traffic searching in a nearby geo area and subscribing to traffic alerts from a particular geo location. A screencast of the system demonstration is available at WSO2 library [18].

Apart from this web interface, we present accuracy measures of our NLP models which were deployed in CEP tool in tables I, II and III. OpenNLP provides APIs to get common accuracy measures for trained models. Document categorizer model accuracy was calculated manually due to the unavailability of an API.

Responsiveness and performance of the system were tested on a running server instance with a test publisher which

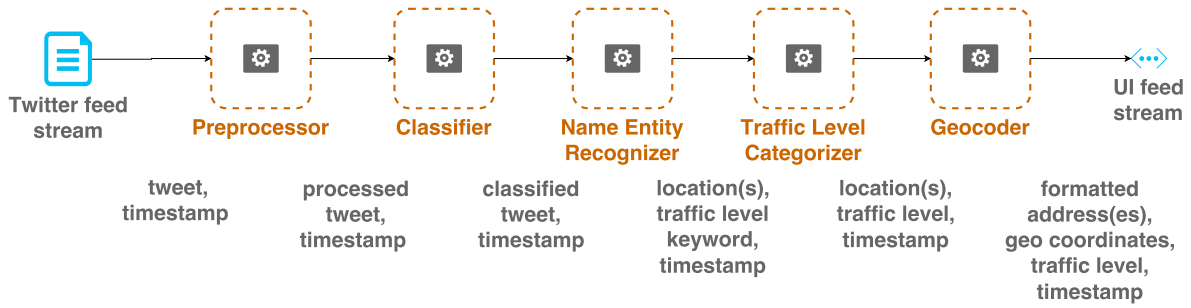

Fig. 2: Pipeline for traffic feed

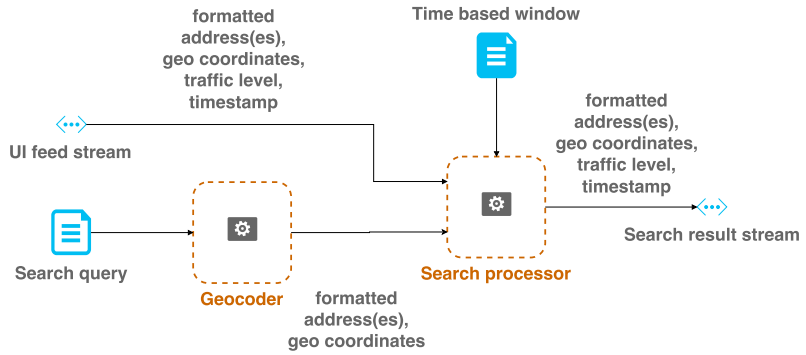

Fig. 3: Pipeline for traffic search

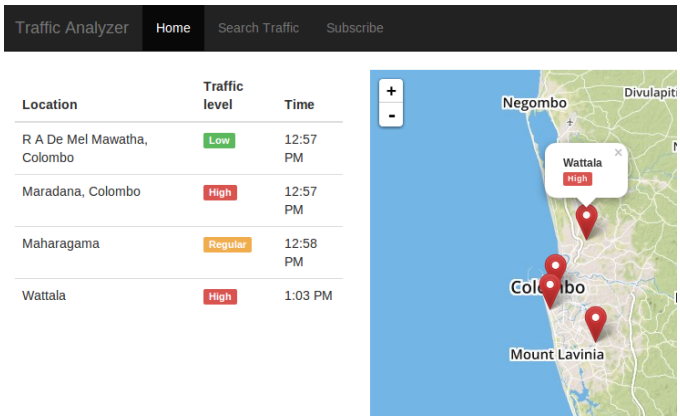

Fig. 5: Traffic feed web interface

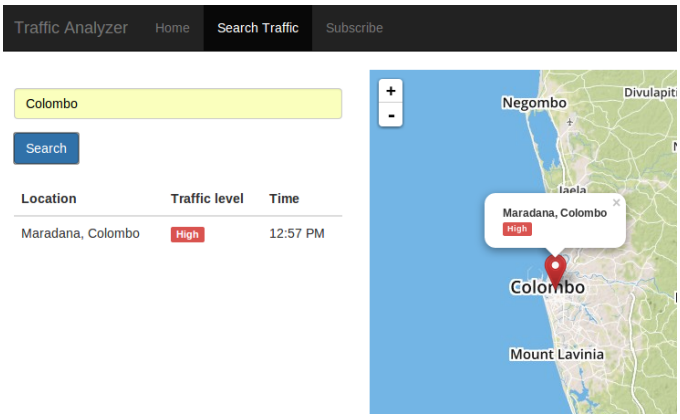

Fig. 6: Traffic search web interface

TABLE I: Performance measures for location NER model

	Cross validation (10%)	Test results
Precision	1.0	0.7452830189
Recall	0.9910714286	0.7596153846
F-measure	0.9955156951	0.7523809524

TABLE II: Performance measures for traffic level NER model

	Cross validation (10%)	Test results
Precision	1.0	0.7083333333
Recall	0.8888888888	0.8095238095
F-measure	0.9411764706	0.7555555556

published tweets to the system from a tweet archive. The system was responsive in total running time and the traffic feed was updated on multiple web interfaces in real-time. Alerts

TABLE III: Performance measures for document categorization model

	Test results
Precision	0.72916667
Recall	0.68627451
F-measure	0.70707071

were generated as email notifications to respective subscribers.

VI. Discussion and Conclusion

In this document we presented a solution to extract useful information from a crowdsourced social networking service by utilizing a NLP/CEP combined approach. Results of this study demonstrate the potential of such model to cope with an application of real-time natural language processing task. Still the scenarios we have demonstrated in this study do not cover all possible use cases of this approach. The model we

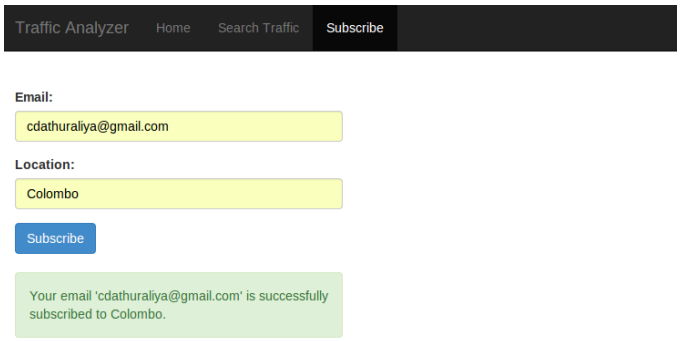

Fig. 7: Subscribe for location web interface

suggest in this study can be extended to tackle any real-time unstructured data stream. Transforming human readable data into machine readable format enables deep processing of data to generate useful information and insights. Further processing may include tasks such as trend analysis, pattern detection and prediction. An immediate extension to this study would be capturing a public feed with a hashtag such as "traffic" to generate worldwide traffic alerts or any other trend worth analysing.

REFERENCES

[1] Multiverse, "Traffic reports for Sri Lanka," Jul. 2015. [Online]. Available: https://road.lk/traffic/

[2] road.lk, "road.lk (@road_lk) | Twitter," Jul. 2015. [Online]. Available: https://twitter.com/road_lk

[3] G. Leduc, "Road Traffic Data: Collection Methods and Applications," Institute for Prospective Technological Studies, Tech. Rep., 2008. [Online]. Available: http://ftp.jrc.es/EURdoc/JRC47967.TN.pdf

[4] R. Schäfer, K. Thiessenhusen, and P. Wagner, "A traffic information system by means of real-time floating-car data," in *ITS world congress*, vol. 2, 2002.

[5] A. Ritter, C. Cherry, and B. Dolan, "Unsupervised modeling of twitter conversations," in *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, ser. HLT '10. Stroudsburg, PA, USA: Association for Computational Linguistics, 2010, pp. 172–180. [Online]. Available: http://dl.acm.org/citation.cfm?id=1857999.1858019

[6] D. Wang, A. Al-Rubaie, J. Davies, and S. Clarke, "Real time road traffic monitoring alert based on incremental learning from tweets," in *Evolving and Autonomous Learning Systems (EALS), 2014 IEEE Symposium on*, Dec 2014, pp. 50–57.

[7] D. Koller, J. Weber, T. Huang, J. Malik, G. Ogasawara, B. Rao, and S. Russell, "Towards robust automatic traffic scene analysis in real-time," in *Decision and Control, 1994., Proceedings of the 33rd IEEE Conference on*, vol. 4, Dec 1994, pp. 3776–3781 vol.4.

[8] Directorate General of Traffic, "Informacin de trfico (Traffic information)," 2015. [Online]. Available: http://infocar.dgt.es/etraffic/

[9] Finnish Transport Agency, "Traffic Situation Service," 2015. [Online]. Available: http://www.finnra.fi/alk/english/

[10] R. Targett, "Twitraffic - UK Traffic Information," 2013. [Online]. Available: http://twitraffic.co.uk/

[11] WSO2, "WSO2 Enterprise Service Bus," 2015. [Online]. Available: http://wso2.com/products/enterprise-service-bus/

[12] The Apache Software Foundation, "Apache OpenNLP," 2010. [Online]. Available: https://opennlp.apache.org/

[13] F. Dellaert, "The Expectation Maximization Algorithm," College of Computing, Georgia Institute of Technology, Tech. Rep., 2002. [Online]. Available: http://www.cc.gatech.edu/~dellaert/em-paper.pdf

[14] WSO2, "WSO2 Complex Event Processor," 2015. [Online]. Available: http://wso2.com/products/complex-event-processor/

[15] S. Suhothayan, K. Gajasinghe, I. Loku Narangoda, S. Chaturanga, S. Perera, and V. Nanayakkara, "Siddhi: A Second Look at Complex Event Processing Architectures," in *Proceedings of the 2011 ACM Workshop on Gateway Computing Environments*, ser. GCE '11. New York, NY, USA: ACM, 2011, pp. 43–50. [Online]. Available: http://doi.acm.org/10.1145/2110486.2110493

[16] WSO2, "Siddhi Language Specification," 2015. [Online]. Available: https://docs.wso2.com/display/CEP310/Siddhi+Language+Specification

[17] WSO2, "Windows - Complex Event Processor 3.1.0 - WSO2 Documentation," 2015. [Online]. Available: https://docs.wso2.com/display/CEP310/Windows

[18] WSO2, "WSO2 CEP Road Traffic Map - An Overview," http://wso2.com/library/demonstrations/2015/02/screencast-wso2-cep-road-traffic-map-an-overview-0/, 2015.