Computers in Industry 78 (2016) 57–69

Contents lists available at ScienceDirect

Computers in Industry

journal homepage: www.elsevier.com/locate/compind

A methodology for traffic-related Twitter messages interpretation

Fábio C. Albuquerque [a], Marco A. Casanova [a], Hélio Lopes [a,*], Luciana R. Redlich [a], José Antonio F. de Macedo [b], Melissa Lemos [c], Marcelo Tilio M. de Carvalho [c], Chiara Renso [d]

[a] Dep. de Informática, Pontifícia Universidade Católica do Rio de Janeiro, Brazil
[b] Dep. de Computação, Universidade Federal do Ceará, Brazil
[c] Instituto TecGraf, Pontifícia Universidade Católica do Rio de Janeiro, Brazil
[d] Inst. di Scienza e Tecn. dellInfo., CNR at Pisa, Italy

ARTICLE INFO

Article history:
Received 1 December 2014
Received in revised form 8 October 2015
Accepted 13 October 2015
Available online 29 October 2015

Keywords:
Twitter analysis
Natural language processing
Traffic monitoring

ABSTRACT

This paper addresses the problem of interpreting tweets that describe traffic-related events and that are distributed by government agencies in charge of road networks or by news agencies. Processing such tweets is of interest for two reasons. First, albeit phrased in natural language, such tweets use a much more regular and well-behaved prose than generic user-generated tweets. This characteristic facilitates automating their interpretation and achieving high precision and recall. Second, government agencies and news agencies use Twitter channels to distribute real-time traffic conditions and to alert drivers about planned changes on the road network and about future events that may affect traffic conditions. Hence, such tweets provide exactly the kind of information that proactive truck fleet monitoring and similar applications require. The main contribution of the paper is an automatic tweet interpretation tool, based on Machine Learning techniques, that achieves good performance for traffic-related tweets distributed by traffic authorities and news agencies. The paper also covers in detail experiments with real traffic-related tweets to access the precision and recall of the tool.

1. Introduction

According to Tennenhouse [1], to be proactive, an application must detect interesting situations before they happen and must be able to handle such situations without human supervision. In particular, to achieve proactive behavior, an application that monitors moving objects must: (1a) model the behavior of the moving objects; (1b) monitor the current state of the objects; (2a) model the environment where the objects move; (2b) monitor the current state of the environment; (3a) detect environment changes that may affect the future behavior of the moving objects; and (3b) adjust the planned behavior of the moving objects to the changes. Inexpensive positioning devices and communication technologies cater for (1b), whereas Web resources, notably Twitter channels, RSS and geoRSS [2] feeds and Open GeoSMS [3], provide valuable data from which to extract event descriptions that affect the environment where the objects move.

In this paper, we focus on the problem of interpreting tweets that describe traffic-related events and that are tweeted by government agencies in charge of road networks and by news agencies. Processing such tweets is of interest for essentially two reasons. First, albeit phrased in natural language, such tweets use a much more regular and well-behaved prose than generic user-generated tweets. This characteristic facilitates automating their interpretation and achieving high precision and recall. Second, government agencies and news agencies use Twitter channels to distribute real-time traffic conditions and to alert drivers about planned changes on the road network and about future events that may affect traffic conditions. Hence, such tweets provide real-time or future information about the road network, which is exactly the kind of information that proactive truck fleet monitoring and similar applications require.

Contributions. The first contribution of the paper is a domain ontology, called TEDO, that models traffic-related situations as *events*, composed of actors, locations and timestamps. The main contribution of the paper is an automatic tweet interpretation tool, based on Machine Learning techniques, that achieves good

* Corresponding author.
E-mail addresses: falbuquerque@inf.puc-rio.br (F.C. Albuquerque), casanova@inf.puc-rio.br (M.A. Casanova), lopes@inf.puc-rio.br (H. Lopes), lrredlich@gmail.com (L.R. Redlich), jose.macedo@lia.ufc.br (J.A.F. de Macedo), melissa@tecgraf.puc-rio.br (M. Lemos), tilio@tecgraf.puc-rio.br (M.T.M. de Carvalho), chiara.renso@isti.cnr.it (C. Renso).

http://dx.doi.org/10.1016/j.compind.2015.10.005
0166-3615/

performance for traffic-related tweets distributed by traffic authorities and news agencies. In particular, given a traffic-related tweet, the tool uses named-entity recognition techniques to identify the location of the event the tweet describes and relation extraction methods to capture relations between the components of the event. The tool transforms each such tweet into a set of RDF triples [4], constructed according to the TEDO ontology. Finally, the paper covers in detail experiments with real traffic-related tweets, which indicate that the tool achieves high precision and recall.

Paper outline. Section 2 introduces the TEDO ontology. Section 3 defines the notions of tagged tweets and dependency trees and illustrates how to represent of a tweet in TEDO. Section 4 describes some implementation details and the experiments with the tool. Section 5 discusses related work. Finally, Section 6 presents the conclusions.

2. TEDO – a traffic event domain ontology

This section introduces the *Traffic Event Domain Ontology*, TEDO, that models traffic-related situations as *events*, composed of actors, locations and timestamps. Section 2.1 briefly reviews some ontology concepts, while Section 2.2 covers the details of the classes and properties of TEDO. Section 3.5 contains an example of a tweet represented in TEDO.

TEDO is generically based on the notion of event [5] and relations between events [6,7]. This design decision reflects the principle that traffic is a process modeled by inter-related discrete events. TEDO also borrows some concepts from two traffic accident ontologies [8,9].

The development of TEDO used two major sources to induce some of the property values. The first source was a gazetteer, which provided the names and coordinates of the locations that populate the class Location (see Section 2.2). The second source was a tweet corpus (see Section 4.2), which induced the values of some of the datatype properties (see Tables 1 and 4).

2.1. A brief review of some ontology concepts

A *class C* is a set of instances or individuals. Rather than referring to "*an instance of class C*" we will simply say "*a c*". For example, instead of "*an instance of class Accident*", we say "*an accident*". We may also declare that a class *D* is a subclass of *C* to indicate that all instances of *D* are also instances of *C*.

We will use XML Schema *simple types*, such as *string, float, boolean and dateTime, enumeration types* that define a list of values and *complex types* created by combining the simple types [10]. We will refer to an XML Schema type simply as a *type*.

A *datatype property P* is a binary relation between the set of instances of a class *D* and the set of values of a type *T*; we say that *D* is the *domain* of *P* and *T* is the *range* of *P*. Datatype properties capture in OWL the equivalent of attributes in UML or in the entity-relationship model, familiar to database designers.

An *object property O* is a binary relation between the set of instances of a class *D* and the set of instances of a class *R*; we say that *D* is the *domain* of *O* and *R* is the *range* of *O*. Object properties capture in OWL the equivalent of relationships in UML or binary relationships in the entity-relationship model. However, OWL does not offer constructs for relations with attributes or *n*-ary relations, with $n > 2$. In such situations, the designer has to resort to the *reification* of the relation [11].

A *cardinality restriction* for a class *C* imposes limitations on the number of occurrences of a datatype or object property *Q* each instance of *C* must have. A *maximum* (or *minimum*) cardinality restriction specifies the maximum (or minimum) number of occurrences of property *Q* each instance of *C* must have. Departing from OWL and adopting UML notation for cardinalities, we use "*m . . n*" to indicate that the number of occurrences of property *Q* for an instance of *C* must be at least *m* and at most *n*. Furthermore, when *m* is "0", *Q* may not be defined for some instances of *C* and, when *n* is "*", *Q* may associate an instance of *C* with an unbounded number of instances of the range of *Q*.

Finally, an *international resource identifier* (IRI) [4] identifies a resource. The notion of IRI is a generalization of URI (Uniform Resource Identifier), allowing non-ASCII characters to be used in the IRI character string. We will store the result of analyzing a tweet as a set of *RDF triples* [4] of the form (s, p, o), where *s* is the *subject*, *p* is the *predicate* and *o* is the *object* of the triple. The subject and the predicate are IRIs and *o* is either an IRI or an XML literal.

2.2. Classes and properties of TEDO

This section describes the classes and properties of TEDO, summarized in Fig. 1 and in Tables 1–3 (the last two also include cardinality restrictions).

Traffic events: Traffic events in TEDO are instances of the class TrafficEvent, which is specialized into the following subclasses (see Fig. 1), with the intended interpretation of their instances:

- Interdiction: an interdiction that affects the traffic;
- Accident: an accident involving one or more vehicles, such as a collision, that affects the traffic;
- Breakdown: a vehicle breakdown that affects the traffic;

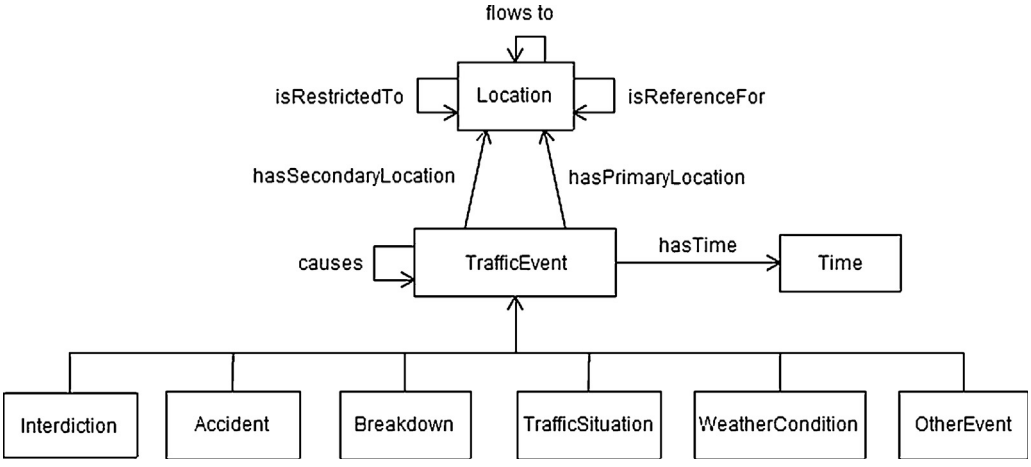

Fig. 1. TEDO classes and object properties (datatype properties omitted for legibility).