Predicting Severity of Road Traffic Congestion using Semantic Web Technologies*

Freddy Lécué, Robert Tucker, Veli Bicer, Pierpaolo Tommasi, Simone Tallevi-Diotallevi, Marco Sbodio

IBM Research, Smarter Cities Technology Centre
Damastown Industrial Estate, Dublin, Ireland
{(firstname.lastname)@ie.ibm.com}

Abstract. Predictive reasoning, or the problem of estimating future observations given some historical information, is an important inference task for obtaining insight on cities and supporting efficient urban planning. This paper, focusing on transportation, presents how severity of road traffic congestion can be predicted using semantic Web technologies. In particular we present a system which integrates numerous sensors (exposing heterogenous, exogenous and raw data streams such as weather information, road works, city events or incidents) to improve accuracy and consistency of traffic congestion prediction. Our prototype of semantics-aware prediction, being used and experimented currently by traffic controllers in Dublin City Ireland, works efficiently with real, live and heterogeneous stream data. The experiments have shown accurate and consistent prediction of road traffic conditions, main benefits of the semantic encoding.

Keywords: #eswc2014Lecue

1 Introduction

As the number of vehicles on the road steadily increases and the expansion of roadways is remained static, congestion in cities became one of the major transportation issues in most industrial countries [1]. Urban traffic costs 5.5 billion hours of travel delay and 2.9 billion gallons of wasted fuel in the USA alone, all at the price of $\$121$ billion. Even worse, the costs of extra time and wasted fuel has quintupled over the past 30 years.

Three ways can be considered to reduce congestion [2]; one is to improve the infrastructure e.g., by increasing the road capacity, but this requires enormous expenditure which is often not viable. Promoting public transport in large cities is another way but it is not always convenient. Another solution is to determine the future states of roads segments, which will support transportation departments and their managers to proactively manage the traffic before congestion is reached e.g., changing traffic light strategy.

Prediction, or the problem of estimating future observations given some historical information, spans many research fields, from Statistics, Signal Processing to Database and Artificial Intelligence. Depending on the level of data representation considered, a prediction problem [3] can be formulated as a standard machine learning classification (for symbolic values) or regression (for numeric values) model [4]. In most of data

* The research leading to these results has received funding from the European Union's Seventh Framework Programme (FP7/2007-2013) under grant agreement ID 318201 (SIMPLI-CITY).

stream mining applications, prediction is estimated by (i) correlating current and past data (e.g., travel times for traffic application), (ii) identifying patterns using different distance metrics [5], and (iii) selecting rules that are used for predicting future conditions [6]. These approaches are designed for very fast processing and mining of (syntactic and numerical) raw data from sensors [7]. They rarely utilize exogenous sources of information for adjusting estimated prediction. Inclement weather condition, a concert event, a car accident, peak hours are examples of external factors that strongly impact traffic flow and congestion [8]. They also all fail in using and interpreting underlying semantics of data, making prediction not as accurate and consistent as it could be, specially when data streams are characterized by texts or sudden changes over time.

We show that the integration of numerous sensors, which expose heterogenous, exogenous, raw data streams such as weather information, road works, city events is a way forward to improve accuracy and consistency of traffic congestion prediction. To this end, we exploit semantic Web technologies and adapt recent research work in semantic predictive reasoning [9] as a way to annotate and interpret semantics of stream data. We extend the latter work by (i) presenting the prediction system[1] and architecture, (ii) focusing on the traffic congestion application, (iii) presenting various technical challenges such as semantic data stream conversion, cross-stream reasoning, consistent prediction, (iv) describing in details all data. As a system-based presentation, this work improves [9] by focusing on InUse criteria i.e., (i) providing technical details of the architecture and implementation, (ii) clearly defining their limitations for further deployments, (iii) drawing new lessons learnt from a more advanced, systemized and inUse prototype, (iv) describing the current interface of the system (Fig.10), (v) reporting new experimental results (Fig.11, Fig.12) against traditional data mining techniques [5]. This work complements [10], which explains traffic congestion in quasi-real-time. In both works data is lifted at semantic level but the diagnosis and predictive approaches are different techniques. Diagnosis is based on semantic matching of events and a probabilistic model while prediction is based on stream auto-correlation, association mining.

This paper is organized as follows: Section 2 presents the Dublin city context and highlights the main challenges we faced to predict the severity of its road traffic congestion. Section 3 describes the system architecture while detailing its limitations. Section 4 reports some experimental results regarding its scalability and accuracy. Section 5 draws some conclusions and talks about future directions.

2 Context: Transportation in Dublin City

2.1 Open Data Sources

All data sources in Table 1 are classified with respect to their velocity i.e., static, quasi stream, stream. They report various types of information coming from static or dynamic sensors, exposed as open, public data and described along heterogenous formats. Quasi stream refers to low throughput sensors. Static sensing refers to stationary platform while dynamic sensing refers to moving objects. The *journey times* data stream is used for (i) monitoring road traffic flow (i.e., free, moderate, heavy, stopped) between static sensors, and (ii) deriving congestion and its severity (i.e., spatial and temporal representation of traffic queues) across 47 routes and its 732 points in Dublin city, all in real-time. Predicting the characteristics of this stream, which we called *main stream*

[1] Prediction part of the live IBM STAR-CITY system (http://dublinked.ie/sandbox/star-city/).

(i.e., stream to be predicted), consists in interpreting, contextualizing and correlating its content with these six exogenous data sources: (1) *road weather condition* which captures specific features of roads conditions e.g., road temperature along 11 static stations, (2) *weather information* e.g., general condition, temperature, precipitation along 19 static stations, (3) *Dublin bus stream* which senses location, speed, delay of 1000 buses every 20 seconds, (4) *social media feeds* which relate traffic-related information e.g., accident, delays, last minute road closure from reputable sources, (5) *road works and maintenance* which plan roads disruptions, their type, duration and (potential) impact on traffic, all updated on a weekly basis, (6) *city events* which characterize social events of various type e.g., music, sport, politics, family, with an average of 187 events per day, all updated on a daily basis.

These data sets have been selected based on their (i) openness, (ii) positive spatial correlation with the *journey times* data stream (i.e., data within a boundary box: max / min latitude: 53.418536 / 53.274247; max / min longitude: -6.095459 / -6.394258), and (iii) factual (positive or negative) impact on traffic flow conditions [8]. Fig.1 spatially represents the static sensors: journey times, road condition, weather stations. The ESRI SHAPE file of Dublin city, spatially describing map-related elements, is used for (i) capturing the shape of roads, and more importantly (ii) identifying nearby roads and their spatial-based segment representation.

Type	Sensing	Data Source	Description	Format	Temporal Frequency (s)	Size per day (GBytes)	Data Provider (all open data)
Stream Data	Static	Journey times across Dublin City (47 routes)	Dublin Traffic Department's TRIPS system[a]	CSV	60	0.1	Dublin City Council via dublinked.ie[b]
		Road Weather Condition (11 stations)		CSV	600	0.1	NRA[c]
		Real-time Weather Information (19 stations)		CSV	[5, 600] (depending on stations)	[0.050, 1.5] (depending on stations)	Wunderground[d]
	Dynamic	Dublin Bus Stream	Vehicle activity (GPS location, line number, delay, stop flag)	SIRI: XML-based[e]	20	4-6	Dublin City Council via dublinked.ie[f]
		Social-Media Related Feeds	Reputable sources of road traffic conditions in Dublin City	Tweets	600	0.001 (approx. 150 tweets per day)	LiveDrive[g] Aaroadwatch[g] GardaTraffic[g]
Quasi Stream	Dynamic	Road Works and Maintenance		PDF	Updated once a week	0.001	Dublin City Council[h]
		Events in Dublin City	Planned events with small attendance	XML	Updated once a day	0.001	Eventbrite[i]
			Planned events with large attendance			0.05	Eventful[i]
Static	Static	Dublin City Map (listing of type, junctions, GPS coordinate)		ESRI SHAPE	No	0.1	Open StreetMap[j]

[a] Travel-time Reporting Integrated Performance System - http://www.advantechdesign.com.au/trips
[b] http://dublinked.ie/datastore/datasets/dataset-215.php
[c] NRA - National Roads Authority - http://www.nratraffic.ie/weather
[d] http://www.wunderground.com/weather/api/
[e] Service Interface for Real Time Information - http://siri.org.uk
[f] http://dublinked.com/datastore/datasets/dataset-289.php
[g] https://twitter.com/LiveDrive - https://twitter.com/aaroadwatch - https://twitter.com/GardaTraffic
[h] http://www.dublincity.ie/RoadsandTraffic/ScheduledDisruptions/Documents/TrafficNews.pdf
[i] https://www.eventbrite.com/api - http://api.eventful.com
[j] http://download.geofabrik.de/europe/ireland-and-northern-ireland.html

Table 1. (Raw) Data Sources for Dublin City Traffic Prediction Scenario.

2.2 Semantic Predictive Reasoning: Research and In Use Challenges

Semantic predictive reasoning [9] is the inference task of *interpreting* and *mining* all *relevant* exogenous streams and their *evolution* through their *temporal changes* and *correlation*. Applied and interpreted in our transportation context, predicting severity

of road traffic congestion consists of three high level challenges:

(C_1) **Handling data variety (csv, xml, tweets, pdf) and velocity (static, stream):** Once exogenous heterogenous data streams are identified as relevant sources for prediction [8], how to represent them in a unified and common model? Which level of expressivity is required? How to automatically extract knowledge from any unstructured data sources, especially streams and social media feeds? How to capture temporal evolution of streams and its underlying knowledge? How to discretize numerical values from streams? For example, how traffic-related social media feeds can be classified around key concepts such as *incident*, *truck accident*, *car delay*, and then spatially and temporally linked with discretized road weather condition, all in real-time?

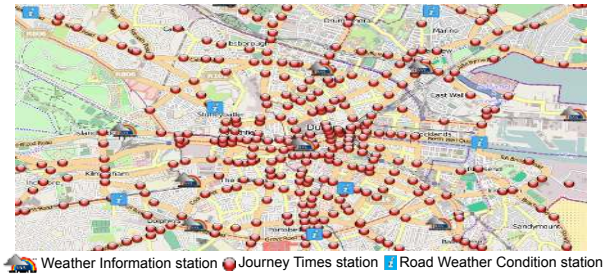


Fig. 1. Spatial Visualization of Static Dublin City Traffic-related Sensors (color print).

(C_2) **Reasoning on the evolution of multiple data streams:** How to understand knowledge evolution and changes of multiple streams on a time basis? How to detect spatial, temporal, semantic correlation in a stream? How to identify associations of streams? E.g., how weather condition is evolving? What were the past time slots with similar conditions? How an *incident-weather* context can be evaluated against historical and real-time traffic flow? Is there any road where a *truck accident* and an *inclement weather condition* could be associated to derive a heavy traffic flow?

(C_3) **Scalable and consistent prediction:** How to rank and select relevant associations of streams in a scalable way? How to use them for achieving consistent prediction? E.g., are there many roads where *truck accident* and inclement weather condition are associated with a heavy traffic flow? Is there more evidence of the latter association when a *car accident* occurred? Will the prediction be consistent with the traffic flow of connected roads?

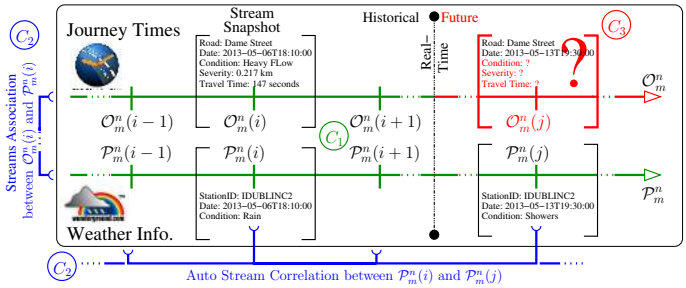


Fig. 2. Articulation of Challenges $C_{i,1\leq i\leq 3}$ in Traffic-related Predictive Reasoning (color print).

On the one hand (C_1), intensively investigated by the semantic Web [11, 12] community to represent and interpret data, can be addressed by mature technologies but still requires some technical adaptations, specially in our streams-based context. On the other hand (C_2) and (C_3) are more recent research challenges which are critical for associating and predicting [9] streams of semantic data. Fig.2 articulates and illustrates these challenges in a simple context of *"predicting the journey times $\mathcal{O}_m^n$ stream, given the exogenous weather information stream $\mathcal{P}_m^n$, where both are evolving on a time basis i.e., from time m to n"*. $\mathcal{O}_m^n(i)$ is called a snapshot of stream $\mathcal{O}_m^n$ at time $i \in [m, n]$. This illustration captures (i) records along one weather station (or stream), (ii) travel condition between two sensors on *Dame Street* at times i, j. We will consider *journey times* as the main stream to be predicted, while the remaining streams from Table 1 are exogenous streams, all used for contextualization (C_1), correlation (C_2) and prediction (C_3). This work focused on addressing these challenges by applying semantic Web related technologies in the context of road traffic congestion prediction.

3 System Architecture for Traffic-related Predictive Reasoning

We report the system architecture (Fig.3) and provide (i) details of all components, (ii) justification of their conceptual and technical specification (if relevant), (iii) limitations and (iv) scalability i.e., applicability to other domains.

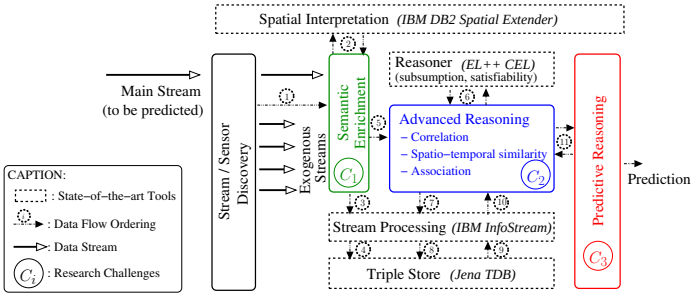


Fig. 3. High-Level System Architecture for Predictive Reasoning (color print).

3.1 High-Level System Architecture

In addition to the components that address all challenges $C_{i,1 \leq i \leq 3}$ in Section 2.2 (described in details in the remaining sections), the system architecture consists of:

• **A spatial interpreter**, required for (i) geocoding some data sources e.g., social media feeds and road works (which only give road identification), (ii) evaluating distance between spatial data but most importantly (iii) retrieving connected roads in Dublin city. IBM DB2 Spatial Extender[2] is used and configured with Ireland SHAPE file[m] in Table 1, and `DB2SE_IRELAND_GEOCODER` geocoder. One of its main strength is its spatial grid index which ensures good query performance i.e., on average 325 ms per request.

• **A stream processing engine**, required for processing data streams e.g., serving real-time semantic streams, materializing knowledge over multiple semantic streams in real-time. IBM InfoSphere Streams[3] is coupled with the *semantic enrichment* and *reasoner*.

[2] `http://www-03.ibm.com/software/products/us/en/db2spaext/`
[3] `http://www-01.ibm.com/software/data/infosphere/streams/`

• **A reasoner**, required for interpreting semantic streams e.g., checking consistency of one or multiple stream(s) at a specific point of time, evaluating subsumption and satisfiability, identifying ABox entailments (i.e., assignment of data instances to concept description based on their representation). CEL DL (Description Logic) reasoner[4] [13] is used as it provides core inference tasks over DL $\mathcal{EL}^{++}$ representations (see justification of this DL family in Section 3.2), implementing a polynomial-time algorithm.
• **A triple store**, for storing the semantic representation of raw data and easily retrieving historical triples. The current prototype uses Jena TDB[5] as RDF store. We preferred the B+ Trees indexing structures which scale better in our context of many (stream) updates.

3.2 Handling Data Variety and Velocity (C_1)

Relevance: On the one hand all of our data is exposed through different formats, which limits not only their integration and semantic interpretation but also any kind of basic inference across data sources. How to measure the similarity of events or road condition? How to classify impact of weather condition on road traffic flow? These are examples of inference problems that need answers for predicting knowledge. By deriving similarity, correlation, association rules we aim at deriving knowledge facts that can be used at prediction time. Such problem cannot be achieved without a minimum of semantic representation. On the other hand data is exposed through (human or device-based) sensors, it is then crucial that real-time semantic conversion can be supported.

Conceptual and Technical Specification: The model we consider to represent static background knowledge and semantics of data stream is provided by an ontology, encoded in OWL 2 EL[6]. The selection of the W3C standard OWL 2 EL profile has been guided by (i) the expressivity which was required to model semantics of data in Table 1, (ii) the scalability of the underlying basic reasoning mechanisms we needed e.g., subsumption in OWL 2 EL is in PTIME [13]. The DL $\mathcal{EL}^{++}$ [14] is the logic underpinning OWL 2 EL and the basis of many more expressive DL. For the sake of readability we illustrate semantic representations using the DL formalism. Fig.4 illustrates a DL sample of the static background knowledge for modeling *journey times* data.

$$Road \sqcap \exists hasTravelTimeStatus.HeavyTrafficFlow \sqsubseteq CongestedRoad \quad (1)$$
$$Road \sqcap \exists hasTravelTimeStatus.StoppedTrafficFlow \sqsubseteq CongestedRoad \quad (2)$$
$$Road \sqcap \exists hasTravelTimeStatus.LightTrafficFlow \sqsubseteq FreeRoad \quad (3)$$
$$CongestedRoad \sqcap FreeRoad \sqsubseteq \bot \ \% \text{ Incompatibility} \quad (4)$$
$$\{r_1\} \sqsubseteq Road \ \% \text{ Individual Definition of a road} r_1 \quad (5)$$

Fig. 4. (Sample) Static Background Knowledge $\mathcal{T}$ for *Journey Times* Data Stream.

We represent knowledge evolution by dynamic and evolutive versions of ontologies i.e., ontology stream [15]. We considered an ontology stream as a sequence of ontologies where each ontology captures a snapshot of a stream at a given point of time t. Fig.5 illustrates a stream snapshot $\mathcal{O}_m^n(t_1)$ i.e., the travel flow severity of a road r_1 from sensor *TRIPS-DCC-44* to *TRIPS-DCC-351*, updated every 1 minute through the *journey times* data stream $\mathcal{O}_m^n$ at date and time t_1 : *2013-04-22T23:01:00*. The ontological representation is important to (i) capture the temporal evolution of its knowledge, (ii) reason across data streams, while the (basic) temporal representation is used to aggregate multiple data sources on a time basis.

[4] http://lat.inf.tu-dresden.de/systems/cel
[5] http://jena.apache.org/documentation/tdb/index.html
[6] http://www.w3.org/TR/owl2-profiles/

$$\mathcal{O}_m^n(t_1) : TravelTimeReport \sqcap \quad (6)$$
$$\exists createdAt.(TemporalEntity \sqcap (\exists inXSDDateTime.\{2013\text{-}04\text{-}22T23{:}01{:}00\})) \sqcap \quad (7)$$
$$\exists reportsForTimeInterval.(\exists hasDurationDescription.(\exists minutes.\{1\})) \sqcap \quad (8)$$
$$\exists hasSourceFrom.\{TRIPS\text{-}DCC\text{-}44\} \sqcap \exists hasSourceTo.\{TRIPS\text{-}DCC\text{-}351\} \sqcap \quad (9)$$
$$\exists reportsObservation.(\{r_1\} \sqcap \exists hasTravelTimeStatus.HeavyTrafficFlow) \quad (10)$$

Fig. 5. (Sample) *Journey Times* Ontology Stream $\mathcal{O}_m^n$ at time *2013-04-22T23:01:00*.

Description: Fig.6 describes the architecture for generating OWL EL ontology streams from raw CSV, tweets, XML, PDF data, all accessed through different mechanisms.

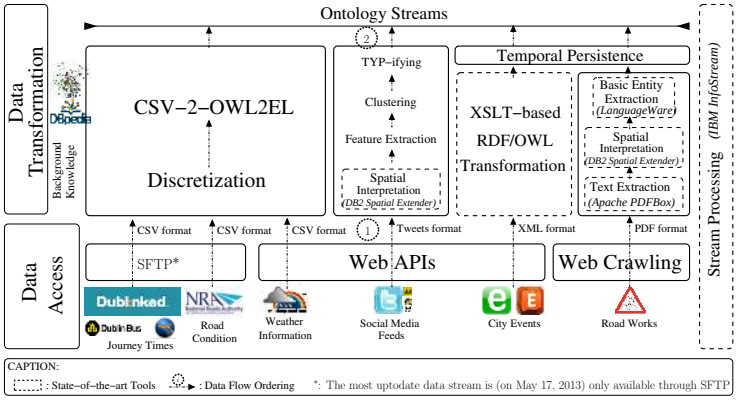


Fig. 6. Semantic Stream Enrichment (color print).

All the ontology streams have the same static background knowledge to capture time (W3C Time Ontology[7] are used for representing (7), (8)), space (W3C Geo Ontology[8] for encoding location of (9)) but differ only in some domain-related vocabularies e.g., traffic flow type, weather phenomenon, event type. These ontologies have been mainly used for enriching raw data, facilitating its integration, comparison, and matching. The DBpedia vocabulary has been used for cross-referencing entities (not described here). We did not make use of the Semantic Sensor Network ontology [9] as it is mainly designed for reasoning over sensor-related descriptions rather than its data and associated phenomenons. In all cases [a,b,c,d], we serve real-time ontology streams by using IBM InfoSphere Streams, where different mapping techniques[15] are used depending on the data format. The main benefits of packaging our approach using stream processing are: (i) easy synchronization of streams (with different frequency updates) and their OWL2 EL transformation, (ii) flexible and scalable composition of stream operations (e.g., transformation, aggregation, filtering) by adjusting its processing units, (iii) identification of patterns and rules over different time windows (Section 3.4), (iv) possible extension to higher throughput sensors. All points are all natively supported by stream processing engines. The following refers to the four types of raw data we consider:

[7] http://www.w3.org/TR/owl-time/
[8] http://www.w3.org/2003/01/geo/
[9] http://www.w3.org/2005/Incubator/ssn/

(a) CSV: A large portion of CSV raw data, exposed by our city sensors, refers to continuous values (e.g., *journey times*). A first step of discretization was required e.g., *Free*, *Moderate*, *Heavy*, *Stopped* traffic flow to conceptualize *journey times*. To this end we evaluated historical data over a period of 6 months to estimate the relevant intervals of values and then associate its concepts. We also adapt existing domain ontologies (e.g., SWEET[10] for (road) weather phenomenon, SIRI-BUS [10] for bus data) and design new vocabularies (e.g., journey times) to cover unsupported descriptions (e.g., travelTimeStatus from/to sensors). Each CSV row is interpreted by a mapping process, handled by our stream processing engine. Fig.7 illustrates the mapping file used for enriching a raw *journey times* data record *[Route: 6, Link: 5, STT: 32, TCS1:44, TCS2: 351]* (collected at timestamp: 1366671660, updated every minute) in its semantic representation (Fig.5) using the static background knowledge. We encoded static city sensors (e.g., *TCS1: 44*) as OWL individual (e.g., *TRIPS-DCC-44*) to reduce the size of the stream description.

```
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix ttr: <http://www.ibm.com/SCTC/ontology/TravelTimeOntology#> .

_:{$uuid}_0 rdf#type ttr#TravelTimeReport .  # $uuid: URI for new travel time report
_:{$uuid}_0 owl#intersectionOf _:{$uuid}_1 . # Each report: intersection of concepts (Fig.5)
_:{$uuid}_1 rdf#first _:{$uuid}_2 .          # Join to the first existential restriction in (9)
_:{$uuid}_2 rdf#type owl#Restriction .       # Existential restriction in (9)
_:{$uuid}_2 owl#onProperty ttr#hasSourceFrom # hasSourceFrom property in (9)
_:{$uuid}_2 owl#hasValue ttr#{$sourceFom} .  # Capture of $sourceFom variable in CSV
_:{$uuid}_1 rdf#rest _:{$uuid}_3 .           # Right part of the Intersection in (9)
_:{$uuid}_3 rdf#first _:{$uuid}_4 .          # Join to the second existential restriction in (9)
_:{$uuid}_4 rdf#type owl#Restriction .       # Another existential restriction in (9)
_:{$uuid}_4 owl#onProperty ttr#hasSourceTo   # hasSourceTo property in (9)
_:{$uuid}_4 owl#hasValue ttr#{$sourceTo} .   # Capture of $sourceTo variable in CSV
_:{$uuid}_3 rdf#rest _:{$uuid}_5 .           # Remaining parts of the Intersection for (7-10)
```

Fig. 7. (Sample) CSV-2-OWL2EL Mapping File for Enriching a *Journey Times* CSV Row.

(b) XML: XML based city events are converted in RDF through an XSL Transformation[11]. Besides updating their representations, it also upgrades their descriptions following [10]. Existing vocabularies such as DBpedia have been used for (i) annotating predefined types of events e.g., capacity, category and (ii) handling basic comparison of events through generalization/specialization. All events are updated only on a daily basis but persist or repeat over time. We simulate these temporal persistence and repetition in the ontology stream by defining their time interval through the *ProperInterval* concept in the W3C Time ontology (by adapting (8)).

(c) PDF: The extraction of the PDF-based road works together with their location, time interval, description and traffic impact is achieved through state-of-the-art tools i.e., (i) PDFBox[12] for extracting text from PDF, (ii) DB2 Spatial extender for geocoding and (iii) LanguageWare[13] for entity extraction through semantic understanding of content. External vocabulary such as DBPedia has been used for type-ing events e.g., road works (e.g., http://dbpedia.org/resource/Roadworks), which ensures potential reuse in the LOD context. The temporal persistence is achieved similarly as city events.

[10] Semantic Web for Earth, Environmental Terminology - `http://sweet.jpl.nasa.gov/`
[11] `http://www.w3.org/TR/xslt`
[12] `http://pdfbox.apache.org/`
[13] `http://www-01.ibm.com/software/globalization/languageware/`

(d) Tweets: Contrary to city events and road works, the semantic enrichment of social media feeds needs a more advanced learning phase. We identify the missing semantics by using an unsupervised learning technique, called Typifier [16], which consists of two major steps, namely *feature extraction* and *clustering*. As a first step, it represents each element (e.g. tweet, event, delay etc.) in the data by a set of features obtained from the attributes i.e., text. E.g., the words such as slow, collision, and delay in the social media feeds can be important features to distinguish its type of *delay*. Once those features are extracted, as a second step, it employs a hierarchical clustering algorithm which aims to maximize intra-cluster homogeneity and inter-cluster separation such that the elements in the same cluster represent the entities of the same type. The clustering method is done automatically. Finally, those clusters are mapped to particular concepts in the background knowledge (DBpedia concepts e.g., Delay, Incident, Accident, Breakdown, Event) in order to enable the semantic lifting of tweets.

Scalability: Our approach can be applicable to any city, and generalized with any other (i) semantic representation e.g., OWL 2 DL, (ii) open (e.g., JSON) or proprietary data format, and (iii) application domains (not only city data) as far as data streams are required (e.g., through sensors). The ontology stream conceptualization also gives the advantage to support real-time querying [17] and reasoning [15] e.g., "*retrieving all roads in Dublin 15 with a heavy traffic flow impacted by inclement weather condition*".

Limitations: The generalization to other domains/cities may require extra manual work to identify, define or extend ontologies. New description of mapping files (e.g., a la CSV-2-OWL2EL or XSLT) would be then required. The entity extraction from natural language (i.e., PDF and tweets) requires some training phases, hence the requirement of some historical data and their pre-processing phases.

3.3 Reasoning on the evolution of multiple data streams (C_2)

Relevance: Once all data in Table 1 is semantically exposed, advanced reasoning techniques are required to capture (i) changes between ontology stream snapshots, and (ii) associations of knowledge at cross-stream level, all on a time basis. The detection of changes supports the understanding of stream evolution, and then provides the basics to compute knowledge auto-correlation along a stream over time. Auto-correlation and association are core reasoning for evaluating potential patterns at one or multi-stream level(s), which are required for predicting severity of congestion. Auto-correlation evaluates semantic similarity of stream snapshots while association aims at deriving rules across streams. E.g., identifying that *"the traffic flow is never stopped on week nights in Dublin 15"* or "*a concert event is always associated with a heavy traffic flow*" are useful facts for prediction purposes.

Conceptual and Technical Specification: On the one hand the TBox (i.e., terminological box containing concepts and their relations) of our static background knowledge, which does not change over time, is classified once using $\mathcal{EL}^{++}$ completion rules [14]. On the other hand the ABox axioms (i.e., relations between individuals and concepts), which are generated by the ontology stream conversion (Fig.6), are internalized into TBox axioms so (i) completion rules can be applied on both axioms, (ii) TBox reasoning (e.g., subsumption, satifiablility) can be performed on internalized ABox axioms.

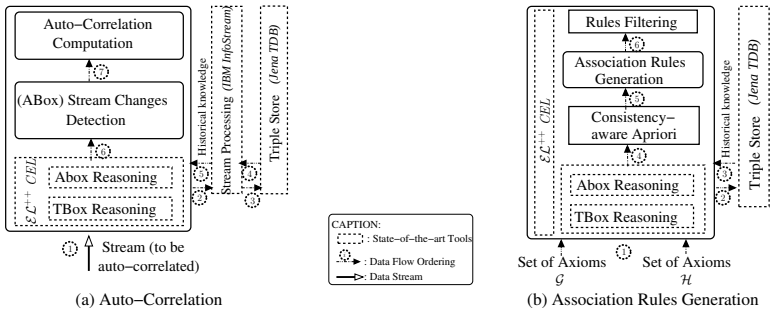
Axiom (11) illustrates some dynamic knowledge at time t_1, as an ABox entailment, derived from axioms (1), (5) in $\mathcal{T}$ (Fig.4) and (10) (Fig.5) using completion rules.

$$\mathcal{T} \cup \mathcal{O}_m^n(t_1) \models^{\text{using axioms (1),(5),(10)}}_{\text{using completion rules in [14]}} \{r_1\} \sqsubseteq CongestedRoad \tag{11}$$

Cross stream association is modeled through DL $\mathcal{EL}^{++}$ rules [18], which extends the DL $\mathcal{EL}^{++}$ expressivity while preserving its polynomial complexity. Intuitively, DL rules are encoded using SWRL rules[14], which is largely based on RuleML. One could, for example, formulate the timeless rule (12) "*the traffic flow of road r_1 is heavy if r_1 is adjacent to a road r_2 where an accident occurs and the humidity is optimum*". This rule connects the *journey times*, *social media* and *weather information* streams.

$$\begin{aligned} HeavyTrafficFlow(s) \leftarrow & Road(r_1) \wedge Road(r_2) \wedge isAdjacentTo(r_1, r_2) \wedge \\ & hasTravelTimeStatus(r_1, s) \wedge hasWeatherPhenomenon(r_1, w) \wedge \\ & OptimunHumidity(w) \wedge hasTrafficPhenomenon(r_2, a) \wedge \\ & RoadTrafficAccident(a) \end{aligned} \tag{12}$$

Description: The auto-correlation of snapshots along an ontology stream is illustrated by (C_2) in Fig.2 and systematized in Fig.8a. We established it by comparing the number of changes i.e., *new*, *obsolete*, *invariant* ABox entailments between snapshots. The number of invariants has a strong and positive influence on auto-correlation. On the contrary, the number of new and obsolete ABox entailments, capturing some differentiators in knowledge evolution, has a negative impact and favors negative auto-correlation. Inconsistencies e.g., (4) are mainly used for capturing incompatible road status at different times e.g., i and j. If captured, they are used to negatively weight the correlation of snapshots $\mathcal{O}_m^n(i)$, $\mathcal{O}_m^n(j)$. i is not an appropriate time to compute the prediction in j.



(a) Auto–Correlation (b) Association Rules Generation

Fig. 8. Stream Auto-Correlation and Association Rules for Prediction.

The generation of association rules (Fig.8b) between streams (and their snapshots) such as (12) is based on a DL extension of Apriori [19], aiming at supporting subsumption for determining association rules. Contrary to the initial version of Apriori, the association is achieved between any ABox elements together with their entailments (e.g., all congested roads, weather, works, incidents, city events delayed buses). The association is possible only in the case their elements appear in at least one point of time of the streams. As the number of rules grows exponentially with the number of ABox elements and entailments in streams, we do not mine all potential rules, but filter them by adapting the definition of *support* (i.e., number of occurrences that support the elements of the rule) and *confidence* (i.e., probability of finding the consequent of the rule

[14] `http://www.w3.org/Submission/SWRL/`

in the streams given the antecedents of the rule) [19] for ontology stream. In addition only consistent associations are considered. For instance $HeavyTrafficFlow(s) \wedge LightTrafficFlow(s)$, which is not consistent with respect to (1), (3), (4), aims at limiting the number of rules to be generated.

Scalability: The approach, systematized from [9] (algorithmic details provided), is generic enough to reason, auto-correlate and cross-associate any ontology streams. Even in the presence of support, confidence and consistency filters, the number of potential rules grows very quickly with the (i) the number of exogenous streams, and (ii) the size of their snapshot. Further investigations along with other metrics are required to reduce this number, that would ensure a better scalability.

Limitations: Jena TDB failed to correctly handle simultaneous updates (coming from various streams). Thus the ontology stream needs to be slightly desynchronized from each other to ensure that Jena TDB handles correctly its transaction model. To this end we simply delayed some of the streams to obtain a sequence of updates instead. We ensure such a desynchronization through our stream processing platform. The B+Trees indexing structure of TDB scales the best in our stream context where large amounts of updates are performed i.e., the transaction model is much better handled in this structure. However there were some scalability issues to handle historical data over more than approximately 110 days. If we do not limit in space and time, and if we do not apply some heuristics (e.g., by restricting to a few days of historic) we could end-up dealing with $1,900,000$+ events (in a - not worst case - context of 458 days of data, where data is updated every 40 seconds). If we consider bus status that is multiplied by $1,000$ i.e., the number of buses. Some challenges such as data / knowledge summarization, stream synchronization are important challenges that need to be tackled, as they both limit the scalability of the approach to some extent.

3.4 Scalable and consistent prediction (C_3)

Relevance: Even if the association rules are filtered by significance (support, confidence) for scalability purpose in Section 3.3, they do not all ensure consistent prediction i.e., prediction which does not contradict other future knowledge facts. Indeed, some rules are specific and may deliver inconsistent prediction. For instance elaborating a prediction with a rule that requires *inclement weather condition* will not be necessarily consistent in a context where the *weather condition is mild*. Towards this issue, rules can be selected based on their applicability in auto-correlated past snapshots, hence reducing the number of rules and ensuring the consistency of their prediction. This motivates why combining auto-correlation and cross-stream association is important.

Description: Fig. 7 presents how auto-correlation is combined with association rule generation for deriving the most relevant rules among streams.

We first identify the context (e.g., *mild weather*, *road closure*) where the prediction is required, and then perform its auto-correlation with historical contexts. Then, we identify and select rules based on their support, confidence and consistency, but only if the consequent of the rule is consistent with the knowledge captured by the exogenous stream. The significance of rules is contextualized and evaluated against only auto-correlated stream snapshots. Thus, the selection of rules [9] is driven by auto-correlation, making the selection knowledge evolution-aware. This ensures to learn rules that could be applied in similar contexts i.e., where knowledge does not drastically change. The prediction can be requested globally to all links of all of the 47 roads

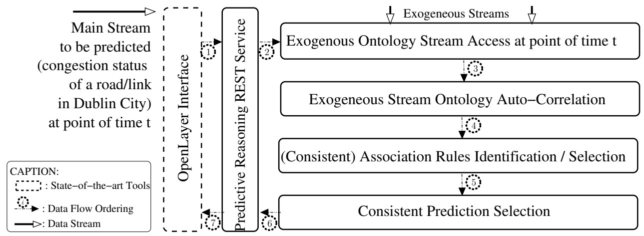


Fig. 9. Scalable and Consistent Prediction.

(red points in Fig.1). Fig.10 reports a 180-minutes ahead prediction of the severity of traffic congestion in Dublin city. The bottom part reports the proportion of free (green), stopped (brown) flow roads of the selected area (top part).

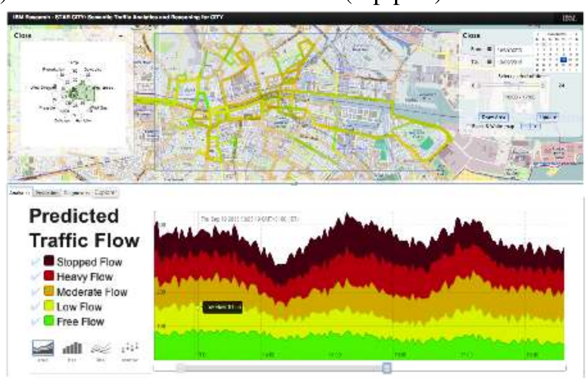


Fig. 10. Traffic Congestion Severity Prediction User Interface (color print).

Scalability and Limitations (cf detailed experimentation in Section 4): Interchanging SWRL rules with SPARQL would benefit the scalability of the approach only when prediction is requested simultaneously. However it would also reduce the expressivity, the number of interesting association rules and the accuracy of prediction.

4 Experimental Results

We focus on the scalability of our approach and its accuracy by (i) comparing its results with a (non semantics) state-of-the-art approach [5] in stream prediction, (ii) analyzing how our approach reacts to the number of stream sources, (iii) reporting the computation time of the various components in Fig.3. Requested by traffic controllers, scalability and accuracy of the system have been extensively tested. The experiments have been conducted on a server of 6 Intel(R) Xeon(R) X5650, 3.46GHz cores, and 6GB RAM.

4.1 Context

Live stream data (Table 1), transformed in OWL/RDF (Table 2) using a static background knowledge (Table 3), are used for experimentation.

The objective is to predict the severity of congestion (i.e., *journey times* stream data) on some Dublin roads in the next hour using exogenous streams. We fixed the size of the stream window to 60 days, which is used for detecting auto-correlation and learning association rules. The impact of the window on predictive reasoning is reported in [9]. Adding more days will slightly increase the accuracy but strongly decrease scalability (because of auto-correlation and rules association generation). The evaluation is achieved on different streams combinations noted [a], [a,b], [a,b,c], [a,b,c,d], [a,b,c,d,e], [a,b,c,d,e,f], [a,b,c,d,e,f,g] in Table 2 to evaluate their impacts on scalability, accuracy.

4.2 Scalability Experimentation and Results

Fig.11 reports the scalability of our approach, noted [14], and compares its computation time with a state-of-the-art approach [5] in stream prediction. Contrary to our

Data Stream	Frequency of Update (s)	Raw Update Size (KB)	Semantic Update Size (KB)	Semantic Update #RDF Triples	Semantic Conversion Computation Time (ms)
[a] Journey Times	60	20.2	6,102	63,000	0.61
[b] Bus	40	66.8	1,766	11,000	0.415
[c] Weather	300	2.2	267	1,140	0.189
[d] Road Works	once a week	146.6	77.9	820	3.988
[e] City Events	once a day	240.7	297	612	1.018
[f] Road Weather	600	715.7	181	660	0.068
[g] Incident	600	0.2	1.0	7	0.002

Table 2. Stream Datasets Details in No Particular Order (average figures).

Ontology	Size (KB)	#Concepts	#Object Properties	#Data Properties	#Individuals	Imported Ontologies	Data Sets Covered
NASA SWEET[12] (IBM adaptation)	158.8	90	40	34	63	W3C Time, Geo	[b,c]
IBM Travel Time	4,194	41	49	22	1,429		[a]
IBM SIRI-BUS	41.9	21	17	18	-		[d]
W3C Time[9]	25.2	12	24	17	14	-	[a-g]
W3C Geo[10]	7.8	2	4	-	-	-	[a-g]
DBpedia	Only a subset is used for annotation i.e., 28 concepts, 9 data properties						[e-g]

Table 3. Static Background Knowledge for Semantic Encoding.

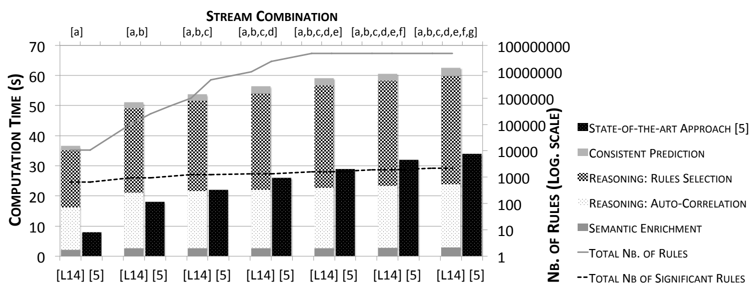


Fig. 11. Scalability of Prediction Computation.

[5]: stream correlation is done at raw data level using (i) statistics-based data analysis and (ii) mathematical properties of the signal (here stream data).

approach scales much better than our approach in all configurations. Our approach requires some (non-negligible) computation time for reasoning on top of the semantics-enriched stream data. The identification of significant rules is strongly impacted by the number of potential rules, which grows exponentially with the number of elements/entailments in streams (secondary vertical axis). Once all rules are identified, consistent prediction is delivered from $1.5s$ to $2.7s$.

4.3 Accuracy Experimentation and Results

Figure 12 reports the prediction accuracy of both approaches. The accuracy is measured by comparing predictions (severity of congestion) with real-time situations in Dublin City, where results can be easily extracted and compared from the raw and semantic data in respectively [5] and our approach. The more the number of streams the better the accuracy of prediction for both approaches. However our approach reaches a better accuracy when text-related streams [d,e,g] are interpreted while the state-of-the-art approach cannot take any benefit of the semantics of such streams. Overall, our approach obtains a better accuracy, mainly because all the rules are pruned based on the consistency of their consequent. By enforcing their consistency, we ensure that rules are selected based on the surrounding context, here exogenous data streams. The semantic

enrichment of data stream is then beneficial for correlating, cross-associating and then predicting streams on a common basis.

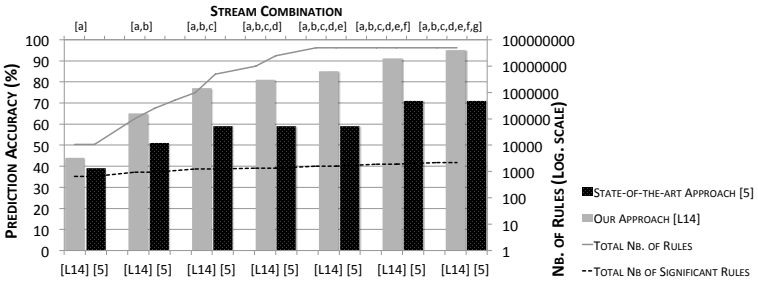


Fig. 12. Accuracy of Prediction.

4.4 Lessons Learned

Our experimental results emphasize the advantage of using semantic Web technologies for predicting knowledge in streams i.e., accuracy, but also point out the scalability limitation, especially compared to pure statistical approaches. The more streams the more rules which positively (resp. negatively) impacts accuracy (resp. scalability). Since state-of-the-art approaches fail to encode text-based streams in pure value-based time series, they simply fail to interpret their semantics. On the contrary, our approach interpret their semantics to enrich the prediction model, ensuring better accuracy.

The reasoning mechanisms in Fig.8 are highly coupled with the polynomial-time CEL reasoner for determining subsumption and consistency, which fits OWL 2 EL. Considering more expressive semantics could have triggered stronger rules while reducing its number, hence improving the scalability (to some extent) and accuracy of prediction. It would also be interesting to evaluate the impact of using a subset of OWL 2 EL on the computation performance and the prediction results. Further experiments are required to provide the most appropriate context and trade-off complexity/expressivity.

In the real world, sensors exhibit noise i.e., they do not observe the world perfectly. The causes range from malfunctioning, mis-calibration, to network issues and attrition breakdown. Noisy data needs to be detected early to avoid a useless semantic enrichment, which could raise to more important problems at reasoning time, reaching to completely inaccurate prediction (due to alteration of rules support and confidence). We partially addressed this problem by integrating some *custom filter operators* at stream processing level to check validity of data e.g., data range checking, exceptions. The integration of new data stream needs a careful analysis of historical data in order to identify the most appropriate filters, avoiding as much noise as possible.

Data streams evolve over time, and release new snapshots at various point of time, making the data stream integration complex. We considered the W3C Time ontology to represent the starting date/time and the duration of each snapshot, but other more complex time feature could have been used e.g., temporal intervals. This would support more complex reasoning to reason over time intervals. For scalability reasons we use basic methods to evaluate loose temporal similarity and then integrate data stream at time level. However research challenges, already tackled by [20], would need to be considered for more accurate temporal joints.

5 Conclusion and Future Work

This work, focusing on transportation, presents how severity of road traffic congestions can be predicted. We (i) presented its challenges, (ii) motivated the use of semantic Web technologies, and (iii) exposed its scalability together with its limitation. We illustrated how recent research work in semantic predictive reasoning, using and interpreting semantics of data, can be exploited, adapted and systematized to ensure accurate and consistent prediction. Our prototype of semantics-aware prediction, experimented in Dublin City, works efficiently with real, live and heterogeneous data stream. The experiments have shown accurate and consistent prediction of road traffic conditions, main benefit of the semantic encoding of information.

As emphasized in Section 4.4, handling (i) noisy data stream, (ii) time reasoning, (iii) flexible stream integration are future domains of investigation. More end-users related evaluations are also planned e.g., user interface, interaction scenarios.

References

1. Schrank, D., Eisele, B.: 2012 urban mobility report. `http://goo.gl/Ke2xU` (2012)
2. Bando, M., Hasebe, K., Nakayama, A., Shibata, A., Sugiyama, Y.: Dynamical model of traffic congestion and numerical simulation. Physical Review E **51** (1995) 1035–1042
3. Han, J., Kamber, M.: Data mining: concepts and techniques. Morgan Kaufmann (2006)
4. Wang, H., Fan, W., Yu, P.S., Han, J.: Mining concept-drifting data streams using ensemble classifiers. In: KDD. (2003) 226–235
5. Papadimitriou, S., Sun, J., Faloutsos, C.: Streaming pattern discovery in multiple time-series. In: VLDB. (2005) 697–708
6. Schrader, C.C., Kornhauser, A.L., Friese, L.M.: Using historical information in forecasting travel times. Transportation Research Board **51** (2004) 1035–1042
7. Min, W., Wynter, L.: Real-time road traffic prediction with spatio-temporal correlations. Transportation Research Part C: Emerging Technologies **19**(4) (2011) 606–616
8. Cairns, S., Hass-Klau, C., Goodwin, P.: Traffic impact of highway capacity reductions: Assessment of the evidence. Landor Publishing (1998)
9. Lécué, F., Pan, J.Z.: Predicting knowledge in an ontology stream. In: IJCAI. (2013)
10. Lécué, F., Schumann, A., Sbodio, M.L.: Applying semantic web technologies for diagnosing road traffic congestions. In: International Semantic Web Conference (2). (2012) 114–130
11. Han, L., Finin, T., Parr, C., Sachs, J., Joshi, A.: Rdf123: from spreadsheets to rdf. In: The Semantic Web-ISWC 2008. Springer (2008) 451–466
12. Abel, F., Gao, Q., Houben, G.J., Tao, K.: Semantic enrichment of twitter posts for user profile construction on the social web. In: The Semanic Web: Research and Applications. Springer (2011) 375–389
13. Baader, F., Lutz, C., Suntisrivaraporn, B.: Cel - a polynomial-time reasoner for life science ontologies. In: IJCAR. (2006) 287–291
14. Baader, F., Brandt, S., Lutz, C.: Pushing the el envelope. In: IJCAI. (2005) 364–369
15. Ren, Y., Pan, J.Z.: Optimising ontology stream reasoning with truth maintenance system. In: CIKM. (2011) 831–836
16. Ma, Y., Tran, T., Bicer, V.: Typifier: Inferring the type semantics of structured data. In: International Conference on Data Engineering (ICDE). (2013) 206–217
17. Calbimonte, J.P., Corcho, Ó., Gray, A.J.G.: Enabling ontology-based access to streaming data sources. In: International Semantic Web Conference (1). (2010) 96–111
18. Krötzsch, M., Rudolph, S., Hitzler, P.: Description logic rules. In: ECAI. (2008) 80–84
19. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules in large databases. In: VLDB. (1994) 487–499
20. Lutz, C.: Interval-based temporal reasoning with general tboxes. In: IJCAI. (2001) 89–96