BeLink: Querying Networks of Facts, Statements and Beliefs

Tien-Duc Cao, Ludivine Duroyon, François Goasdoué, Ioana Manolescu, Xavier Tannier

BeLink: Querying Networks of Facts, Statements and Beliefs

Tien-Duc Cao
Inria and LIX (UMR 7161,
CNRS and Ecole polytechnique)
Palaiseau, France
tien-duc.cao@inria.fr

Ludivine Duroyon
Univ Rennes, CNRS, Inria, IRISA
Lanion, France
ludivine.duroyon@irisa.fr

François Goasdoué
Univ Rennes, CNRS, Inria, IRISA
Lanion, France
fg@irisa.fr

Ioana Manolescu
Inria and LIX (UMR 7161,
CNRS and Ecole polytechnique)
Palaiseau, France
ioana.manolescu@inria.fr

Xavier Tannier
Sorbonne Université and Inria
Paris, France
xavier.tannier@sorbonne-universite.fr

ABSTRACT

An important class of journalistic fact-checking scenarios [2] involves verifying the *claims* and *knowledge* of different actors at different moments in time. Claims may be about *facts*, or about other claims, leading to chains of hearsay. We have recently proposed [4] a data model for (time-anchored) facts, statements and beliefs. It builds upon the W3C's RDF standard for Linked Open Data to describe connections between agents and their statements, and to trace information propagation as agents communicate. We propose to demonstrate BeLink, a prototype capable of storing such interconnected corpora, and answer powerful queries over them relying on SPARQL 1.1. The demo will showcase the exploration of a rich real-data corpus built from Twitter and mainstream media, and interconnected through extraction of statements with their sources, time, and topics.

1 INTRODUCTION

Fact-checking journalists oftentimes need to check *who said what when.* Such an analysis may be made to determine where a public figure up for (re)election stands with respect to a given issue (a famous example is John Kerry's Senate voting history on the war in Irak[1], or the public positions of members of a whole political party on an issue (e.g., the projected Wall between the US and Mexico). Statements are made by individuals or organizations, on certain topics, and typically claim to refer to (real-world) facts. Different actors often make different statements about the same fact or about each other statements. An actor may also make different statements about the same thing at different points in time. Professional standards of journalistic work lead to a high interest in modeling and being able to show statement *sources*, which extends our (informal) definition of data of interest to: *who said what when where.* The source can be public (e.g., a speech whose transcript is available on the Web, or a tweet) or it can be private (e.g., an email that journalists acquire through their sources, or a transcript of a conversation with a source).

Many current tools allow analyzing online media to answer specific questions, for instance, CrowdTangle allows to monitor social media and extract events, Twitonomy and TwitterAnalytics are specifically devoted to analyzing Twitter content etc. We propose to demonstrate BeLink, a tool for extracting and analyzing (timed)

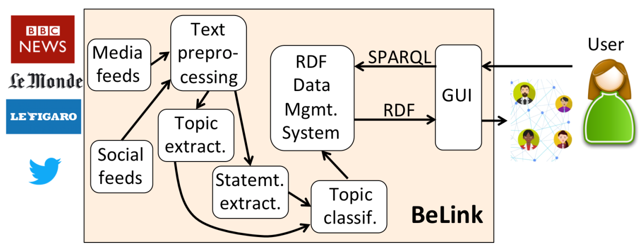

Figure 1: BeLink architecture.

facts, statements and beliefs, out of a set of varied data sources. At the core of BeLink is a *generic data model for real-world facts, statements, and beliefs* we recently introduced [4], including (but not limited to) those expressed on social networks. *Time* plays an important role, since we must capture when events occur (or when facts hold), and when different statements are made about them; this serves, for instance, to track position reversals in time, or to keep track of promises[2]; thus, our model incorporates classical temporal database concepts to attach time to facts, statements, and communications. Further, we take inspiration from classical AI techniques for modeling *agents' beliefs* in order to capture the connections between actors and their statements. The model is based on W3C's Resource Description Framework (RDF) concrete graph data model. This makes instances of our data model *easy to share and combine (link)* with any other RDF dataset, e.g., one that classifies actors according to their political opinions, connections to companies etc., to enable even more analyses. Beyond being "white-box" (as opposed to models not publicly shared, used by existing media analysis tools), the biggest advantage of our model is to be *comprehensive* (modeling all the above aspects: facts, agents, beliefs, and information propagation), *interoperable* (being RDF), *extensible* (other data sources can be turned into instances of our model) and endowed with *formal semantics.* Also, adopting RDF allows us querying instances of our model with *powerful SPARQL 1.1 queries*, notably featuring *property paths*; these capture information propagation along paths of unbound length.

Figure 1 outlines the architecture of BeLink. It comprises a set of *feeds* which gather content from public mainstream and social media; a set of *extractors* to identify statements in this content, to determine their *topics*, the statement time etc. The structured data

[1] http://tiny.cc/92s43y

[2] http://tiny.cc/nys43y

resulting from the extraction is converted into our RDF data model and loaded into an RDF data management system. Users formulate queries over it with a GUI, where they also obtain query results.

Below, Section 2 presents the data model, and Section 3 describes our system. We then outline the demonstration scenario, and finally discuss related works and conclude.

2 RDF DATA MODEL FOR FACTS, STATEMENTS AND BELIEFS

We briefly recall our data model for representing timed facts, beliefs and statements.

2.1 Agents, Time, Facts and Beliefs

Agents are individuals, organizations (companies, media etc.) or other "party" which make statements or learn about them. We model agents as RDF resources of type *Agent*. From now, we use τ as a shorthand for the standard RDF typing property *rdf:type*.

Time is modeled using *time intervals* characterized by a *start* and an *end* time point, both represented using the W3C's XML Schema `dateTime` type[3], under the form YYYY-MM-DD[THH:MM]. The special constants $-\infty$ and $+\infty$ denote interval bounds for assertions that have not changed and will not change respectively, whereas *now* designates the current time. A sample time interval is: (t_0, *begin*, 2018-12-01T09:00), (t_0, *end*, 2018-12-09T19:30).

Facts are postulates about real-life events stored in the database. They are modeled with RDF resources of type *Fact*. In the following, we use F_1, F_2 etc. as fact resources. Each fact has a *time* property specifying when the fact is supposed to occur. Further information about the fact itself is the value of the property *description*; this can be e.g., a text, or an RDF resource having more properties etc. For instance, the fact "Yellow vests[4] protest against fuel prices in Bordeaux, France; on Decembre 1st, 2018" is represented as: (F_1, τ, Fact), (F_1, *description*, d_1), (d_1, *who*, Giletsjaunes), (d_1, *what*, protest), (d_1, *where*, Bordeaux), (F_1, *time*, t_0), (t_0, *begin*, 2018-12-01), (t_0, *end*, 2018-12-01).

Figure 2 sketches the running example developed along Section 2; oval nodes denote URIs, while text nodes correspond to literals. Some information such as edge labels, etc. are ommitted to avoid clutter; URIs representing agents are signaled by a "user" pictogram.

Beliefs relate agents with what they believe; we model them as resources of type *Belief*. We use "believes" to model any among: *has knowledge (is informed) of*, *thinks* or *believes* something etc. A belief can be a positive belief (the agent *does* believe something) or a negative one (the agent *does not* believe it). A belief is characterized by: (*i*) the agent holding the belief, which is the value of a *from* property whose subject is the belief; (*ii*) the time when the agent holds the belief, represented by a *time* property; (*iii*) the belief subject, which is the value of a *believes* property, can be a fact, another belief, or a communication (to be discussed shortly); (*iv*) finally, a *sign* property whose values can be + or −, indicating whether the agent actually believes the subject of the belief, or not. For simplicity, we assume the *sign* property is present only when its value is −; otherwise, we assume its value is +, i.e., the agent does hold the belief. For example, building on the above fact

[3]See https://www.w3.org/TR/xmlschema11-2/#dateTime
[4]"Yellow Vests" or *"Gilets jaunes"* refers to a continuing social movement in France (since November 2018) which has generated a huge amount of social and media content.

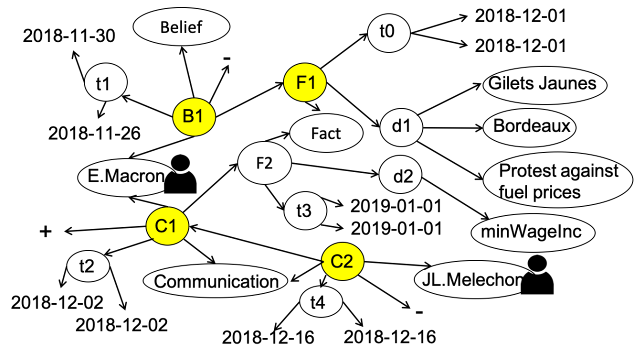

Figure 2: Sample facts and beliefs.

F_1, we represent "E. Macron has believed from November 26th, 2018, to November 30th, 2018, that Yellow vests will not protest against fuel prices in Bordeaux, France; on December 1st, 2018." by: (B_1, τ, Belief), (B_1, *sign*, −), (B_1, *from*, E.Macron), (B_1, *time*, t_1), (t_1, *begin*, 2018-11-26), (t_1, *end*, 2018-11-30), (B_1, *believes*, F_1).

2.2 Belief sharing: communications

Information (beliefs) spread through time through communications. Each communication is characterized by: (*i*) an agent who is the transmitter, indicated by its *from* property; (*ii*) optionally, one or more agents who are the receivers, indicated by the *to* property; (*iii*) a subject, which is the value of the *communicates* property, which can be a fact, a belief or another communication; (*iv*) a sign, whose value is given by a *sign* property + or −, indicates whether the agent actually agrees or disagrees with the subject of the communication; (*v*) a time encoded by the *time* property. If the receiver is not specified, we assume it is a *public* communication. These are considered available to anyone, e.g., anyone can have access to the newspaper, TV, Web source where the communication was made. Communications with one or more specific receivers are considered private; only the transmitter and the receiver are assumed aware of this communication. For example, we represent "On Dec 10th, 2018, E. Macron stated that there will be an increase of the minimum wage on January 1st, 2019." by: (C_1, τ, Communication), (C_1, *communicates*, F_2), (C_1, *from*, E.Macron), (C_1, *sign*, +), (C_1, *time*, t_2), (t_2, *begin*, 2018-12-10), (t_2, *end*, 2018-12-10), (F_2, τ, Fact), (F_2, *description*, d_2), (d_2, *what*, minWageInc), (F_2, *time*, t_3), (t_3, *begin*, 2019-01-01), (t_3, *end*, 2019-01-01).

2.3 Records and databases

The above example shows that some facts are considered to hold in the database, e.g., F_1 in the above example which cannot be disputed hence holds in the database, whereas others are only merely *believed* or *communicated* by some agents, e.g., F_2 above: the minimum wage increase is *just* a fact announced by E. Macron, whether or not it will really happen on the given date. Also, the database may state that an agent *A* holds a belief, or makes a communication, but this is different from the database stating that *according to agent B*, agent *A* believes, respectively communicates it. In the former case, something holds *according to the database*, i.e., is considered an undisputed (checked) fact; in the latter, the database merely states that something holds *according to B*. Both may

Direct	*"Lacking concrete measures, our international meetings become useless and even counterproductive", warned Emmanuel Macron on Thursday"*
Indirect	*"Edouard Philippe announces he is ready to receive representatives of gilets jaunes"*

Figure 3: Sample extracted statements.

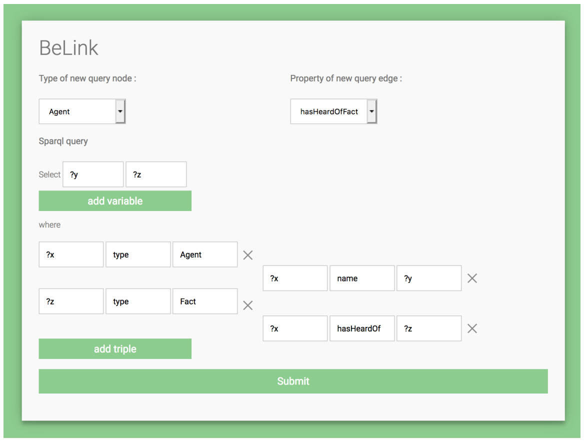

Figure 4: Query formulation GUI.

also coexist in the database, e.g., if Jean-Luc Mélenchon, a political leader of the Opposition, communicates that the above C_1 announcement of E. Macron will not happen: (C_2, τ, Communication), (C_2, *from*, J. − L. Melenchon), (C_2, *sign*, −), (C_2, *communicates*, C_1), (C_2, *time*, t_4),
(t_4, *begin*, 2018-12-16), (t_4, *end*, 2018-12-16).

Records We introduce a special type *Record* which we attach to any fact, belief or communication that *holds according to the database*. Note that each record may be the root of a potentially long chain of beliefs and communications; each such chain ends in a *Fact*[5]. We illustrate this below using our above running example. In Figure 2, nodes of type Record are shown on a yellow background. The fact F_1, belief B_1 and the communication C_1 of E. Macron are of type *Record*, as well as the communication C_2 of JL. Mélenchon. However, F_2 is not of type *Record* because it is only true according to E. Macron (and disputed by JL. Mélenchon).

Database In our model, a *database* is a *set of facts, beliefs and communications*, each may be a record or not, stored as RDF triples as discussed above. Figure 2 displays such a database, in which F_1, B_1, C_1, C_2 are the database records (shown as yellow nodes). We also preserve (not shown) the URI of the original communication (tweet, interview etc.) to allow verifying the information source(s).

Our generic model can be **customized/instantiated** in many different ways to suit different corpora. This is naturally supported by RDF ontologies, which allow e.g. to state that *Journalists* and *Politicians* are subclasses of *Agents*, *statesInArticle* and *saysOnLiveRadio* are subproperties of *communicates* etc. Thus, we describe the data at the most specific level known, and query it either at a specific (e.g., *saysOnLiveLiveRadio*) or at a more generic level (e.g., *communicates*).

3 BELINK COMPONENTS

RDF DBMS Following a recent benchmark of *property path* support [12], a feature introduced in SPARQL 1.1 that we use to ask complex queries of practical interest through our GUI, we use RDF4J (formerly known as Sesame) v2.4.2. as our RDF DBMS (recall the architecture in Figure 1).

Feeds and extractors We use `twint`[6] to get tweets comprising a given keyword. We extract from them the links to media articles they mention, and build our news feed using `News-Please` [9], which returns the bodies of media articles, given their URIs.

We rely on a statement extraction tool[7] we developed to collect *statements* and their authors, i.e., phrases describing someone stating something. Both direct and indirect statements are found, as illustrated in Figure 3, where the author is shown in violet. To extract time information from text, we rely on HeidelTime [15].

We assign *topics* to the various texts, to allow linking their contents. For that purpose:

(*i*) As a pre-processing, we tokenize texts using Spacy[8]. We remove stop words, detect numbers and time values (e.g. "12h30") using simple regular expressions, and replace them respectively by ⟨NUM⟩ and ⟨TIME⟩.

(*ii*) We obtain the topics themselves from the corpus by applying the Scholar topic modeling algorithm [1]. It outputs each topic as *a list of semantically related words*, e.g., a topic defined by {"law enforcement", "police agents", "police brutality"} can be inferred to be about the police.

(*iii*) To relate texts to these topics, we use WeSTClass [14] based on machine learning (ML). An advantage of this tool does not require training data. Instead, given a set of topics (each as a set of keywords), it generates "pseudo documents" using word embeddings for each topic, using the Von Mises-Fisher distribution [7]. Then, it trains a ML classifier on these documents, which is finally used to assign for each text, a *score* (between 0 and 1) for each topic. We consider a text is *about* the three topics with the highest WeSTClass scores. [14] proposed two deep learning models: Convolutional Neural Network (CNN) and Hierarchical Attention Network (HAN) We chose CNN since it yielded the most accurate results; the parameters of our model are at https://tiny.cc/z6s43y. The statement extraction tool is Java-based; the feeds and extractors modules are in Python.

GUI We developed an interactive JavaScript GUI based on jQuery, as Figure 4 shows. It allows users to specify SPARQL queries triple by triple. They can choose the types (Agent, Fact, Belief etc.) of the query variables, and then a context-dependent menu allows them to select how these variables are related to others or to constants, either through properties of our data model or via more complex predicates (e.g. *hasHeardOf*, see below), whose SPARQL 1.1 syntax is hidden to users.

4 DEMONSTRATION SCENARIO

We have built an original **French corpus** for the demo, of independent interest given the rarity of non-English text resources. We

[5]This follows the natural interpretation that any belief or communication carries over something, thus the chain must end in a Fact.
[6]https://github.com/twintproject/twint
[7]https://gitlab.eurecom.fr/asrael/source-extractor
[8]http://spacy.io

collected 96.513 French tweets which contain the words "gilets jaunes". Following links in these tweets, we obtained 56.657 news articles, together with their metadata (title, author and publication time). Thus, we obtain a corpus $C = (C_t, C_s)$, where C_t contains the tweets (12MB), while C_s contains the extracted statements (76.008, together occupying 13MB). Each item from C has a *text* which is the tweet content or an extracted statement, an *author* (the tweet author, respectively, the author of the quote as recognized by `source-extractor`), and the *publication time*.

Topic extraction lead to 14 topics for C_s and 18 topics for C_t (see https://tiny.cc/v9s43y) with which we then annotated C. To evaluate the quality of the annotation, we manually annotated 192 randomly chosen C_s statements, and 162 randomly chosen tweets C_t, and split them equally into development and test sets.

	F1, Dev	F1, Test	MAP@3, Dev	MAP@3, Test
C_s	0.727	0.711	0.738	0.721
C_t	0.785	0.817	0.823	0.842

We have measured the F1 score (based on recall and precision) and the mean average precision (MAP@3), a measure commonly used for list similarity. These measures (above) are quite high, confirming the quality of our extraction. Then, for each c $\in C$ whose publication time is t, authored by A, and for which O is among the three highest-score topics, we create a Communication Com and add the following RDF snippet to the database: (Com, from, A), (Com, communicates, F), (Com, time, t), (F, description, D), (D, hasTopic, O), (D, hasText, c). Tweets which cite/retweet one another lead to *chains of communications* in our model (Section 2); their authors, or people mentioned in them, as well as their topics, allow interconnecting C.

Queries Users will be able to query the instance, e.g., to find how members of opposing political parties communicate on a given topic, or how someone's statements vary from statements other people attribute to him. An interesting family of queries trace the *propagation* of information across chains of beliefs and communications; this translates in SPARQL property paths. They can be understood based on the generic query Q_0 shown below.

$Q_0(?a, ?c, ?b, ?e, ?s) \leftarrow$ (?c, τ, Record),
(?c, (from|to), ?a),
(?c, (believes|communicates)+, ?s)
(?c, time, ?t), (?t, begin, ?b), (?t, end, ?e)

Q_0 captures who has heard of what, when, and how. We consider an agent *hears about* something when either the agents believes it, or the agent is the recipient of a communication about it. Q_0 returns: the agent a, the communication c, its time (as an interval spanning from b to e), and its subject s. Note the regular path expression (of potentially unbounded length) going from c to the subject s: it captures the propagation of hearsay, i.e., if c is about a belief b (or another communication c') which is about event ev, then Q_0 returns t together with a and c, since it is through c that a heard of ev.

For example, we can specialize Q_0 in a query asking who has heard of a topic *to*: just bind the subject ?*s* to *to*. A similar query gives profile of the agents who have heard of *to*; we can aggregate them along their political affiliation, with the help of a small knowledge built in our collaboration with Le Monde's fact-checkers. We can search which agent (political party) first mentions a topic and how long it takes the opposing party to react, find the dominant topic in a time period, etc. Users will compose such queries through the GUI, in which the property path corresponding to *hasHeardOf* is available like a "property label" they can select.

5 RELATED WORK AND CONCLUSION

Modelling facts, statements and beliefs to further search and analyze them has recently gained interest in computational journalism and fact-checking, e.g., monitoring sources, extracting claims, checking them w.r.t. reference sources and publishing obtained results [2].

Our work combines database-style modelling with extraction to produce interesting, linked instances of timed facts, statements and beliefs, which we exploit through complex SPARQL 1.1 queries of practical interest in a data journalism setting, posed via a form-based GUI. *Relational belief databases* [5] represent (i) facts and (ii) positive or negative beliefs of agents on facts, or on beliefs. We do not consider belief inference; instead, we focus on storing information, and how it propagates between agents.

Our time representation borrows from *temporal databases*, studied first in a relational setting [13] then for RDF. [8] attach time points or time intervals to triples; [11] allows intervals with unknown bounds on which Allen's constraints can be set.

RDF has also been used in fact-checking. In [3], claims are RDF triples checked against a knowledge base such as DBPedia. Fact-Minder [6] links the entities found in documents (web pages, etc.), to their descriptions in a knowledge graph, to guide manual fact-checks. Finally, [10] proposes a complete fact-checking systems, from claim extraction to analysis and publication of the results; it check claims against several knowledge bases. However, it does not model facts, beliefs, and their propagation.

REFERENCES

[1] Dallas Card, Chenhao Tan, and Noah A. Smith. 2018. Neural Models for Documents with Metadata. *ACL* (2018).
[2] Sylvie Cazalens, Philippe Lamarre, Julien Leblay, Ioana Manolescu, and Xavier Tannier. 2018. A Content Management Perspective on Fact-Checking. In *WWW*.
[3] Giovanni Luca Ciampaglia, Prashant Shiralkar, Luis M. Rocha, Johan Bollen, Filippo Menczer, and Alessandro Flammini. 2015. Computational fact checking from knowledge networks. *PloS one* 10, 6 (2015).
[4] Ludivine Duroyon, François Goasdoué, and Ioana Manolescu. 2019. A Linked Data Model for Facts, Statements and Beliefs. In *Int'l. Workshop on Misinformation, Computational Fact-Checking and Credible Web*. https://hal.inria.fr/hal-02057980
[5] Wolfgang Gatterbauer, Magdalena Balazinska, Nodira Khoussainova, and Dan Suciu. 2009. Believe It or Not: Adding Belief Annotations to Databases. *PVLDB* (2009).
[6] François Goasdoué, Konstantinos Karanasos, Yannis Katsis, Julien Leblay, Ioana Manolescu, and Stamatis Zampetakis. 2013. Fact checking and analyzing the web. In *SIGMOD*.
[7] Siddharth Gopal and Yiming Yang. 2014. Von Mises-Fisher Clustering Models. In *ICML (ICML'14)*. JMLR.org.
[8] Claudio Gutiérrez, Carlos A. Hurtado, and Alejandro A. Vaisman. 2005. Temporal RDF. In *ESWC*.
[9] Felix Hamborg, Norman Meuschke, Corinna Breitinger, and Bela Gipp. 2017. news-please: A Generic News Crawler and Extractor. In *Int'l Symposium of Information Science*.
[10] Naeemul Hassan, Gensheng Zhang, Fatma Arslan, Josue Caraballo, Damian Jimenez, Siddhant Gawsane, Shohedul Hasan, Minumol Joseph, Aaditya Kulkarni, Anil Kumar Nayak, et al. 2017. ClaimBuster: The First-ever End-to-end Fact-checking System. *PVLDB* 10, 7 (2017).
[11] Carlos A. Hurtado and Alejandro A. Vaisman. 2006. Reasoning with Temporal Constraints in RDF. In *Principles and Practice of Semantic Web Reasoning (PPSWR)*.
[12] Daniel Janke, Adrian Skubella, and Steffen Staab. 2017. Evaluating SPARQL 1.1 Property Path Support. In *Int'l Workshop on Benchmarking Linked Data*.
[13] Christian S. Jensen and Richard T. Snodgrass. 1999. Temporal Data Management. *IEEE TKDE* 11, 1 (1999).
[14] Yu Meng, Jiaming Shen, Chao Zhang, and Jiawei Han. 2018. Weakly-Supervised Neural Text Classification. *CIKM* (2018). arXiv:1809.01478v2
[15] Jannik Strötgen and Michael Gertz. 2010. HeidelTime: High Quality Rule-Based Extraction and Normalization of Temporal Expressions. In *Semantic Evaluation*.