DeFacto - A Multilingual Fact Validation Interface

defacto.aksw.org

René Speck, Diego Esteves, Jens Lehmann, and Axel-Cyrille Ngonga Ngomo

University Leipzig, Institute for Computer Science, Agile Knowledge Engineering and Semantic Web (AKSW), D-04009 Leipzig, Germany
email: {lastname}@informatik.uni-leipzig.de

Abstract. The curation of a knowledge base is a key task for ensuring the correctness and traceability of the knowledge provided in the said knowledge. This task is often carried out manually by human curators, who attempt to provide reliable facts and their respective sources in a three-step process: issuing appropriate keyword queries for the fact to check using standard search engines, retrieving potentially relevant documents and screening those documents for relevant content. However, this process is very time-consuming, mainly due to the human curators having to scrutinize the web pages retrieved by search engines. This demo paper demonstrate the RESTful implementation for DeFacto (Deep Fact Validation) – an approach able to validate facts in RDF by finding trustworthy sources for them on the Web. DeFacto aims to support the validation of facts by supplying the user with (1) relevant excerpts of web pages as well as (2) useful additional information including (3) a score for the confidence DeFacto has in the correctness of the input fact. To achieve this goal, DeFacto collects and combines evidence from web pages written in several languages. We also provide an extension for finding similar resources obtained from the Linked Data, using the *sameas.org* service as backend. In addition, DeFacto provides support for facts with a temporal scope, i.e., it can estimate the time frame within which a fact was valid.

1 Introduction

The evolution from an industrial society to an information and knowledge society has a direct impact on the manner we search, access and share information. Within this context, knowledge bases play an important role in various communities worldwide. Hence, the assessment of given knowledge is of high importance. According to Zaveri [1], one important aspect of ensuring knowledge quality is to provide users with the provenance of facts included therein. However, only a small fraction of the knowledge bases on the Linked Data Cloud currently provide provenance information. As reported by Sindice[1], less then 10% of the RDF documents indexed contain metadada derived from trendy provenance vocabularies, such as `dct:creator`, `dct:created`, `dct:modified`, `dct:contributor` and `dct:source`[2]. The lack of provenance makes the verification of the facts in such knowledge bases extremely tedious. In this paper, we present a RESTful implementation for DeFacto[2,3], a framework for fact

[1] `http://www.sindice.com`
[2] `http://dublincore.org/documents/dcmi-terms/`

```
@prefix fbase: <http://rdf.freebase.com/ns/> .
@prefix rdfs:  <http://www.w3.org/2000/01/rdf-schema#> .
@prefix dbo:   <http://dbpedia.org/ontology/> .
@prefix dbr:   <http://dbpedia.org/resource/> .
@prefix fr-dbr: <http://fr.dbpedia.org/resource/> .
@prefix de-dbr: <http://de.dbpedia.org/resource/> .
@prefix owl:   <http://www.w3.org/2002/07/owl#> .
@prefix xsd:   <http://www.w3.org/2001/XMLSchema#> .
@prefix skos:  <http://www.w3.org/2004/02/skos/core#> .

fbase:m.0dt39
  rdfs:label    "Nobel Prize in Physics"@en, "Prix Nobel de
      physique"@fr, "Nobelpreis für Physik"@de ;
  owl:sameAs    fr-dbr:Prix_Nobel_de_physique , de-dbr:
      Nobelpreis_für_Physik , dbr:Nobel_Prize_in_Physics ;
  skos:altLabel "Nobel Physics Prize"@en , "Nobel laureates in
      physics"@fr , "Physik-Nobelpreis"@de ...

fbase:m.0jcx__24
  dbo:award  fbase:m.0dt39 ;
  dbo:startYear   "1921"^^xsd:gYear ;
  dbo:endYear     "1921"^^xsd:gYear .

fbase:m.0jcx
  rdfs:label         "Albert Einstein"@fr , "Albert Einstein"@en
       , "Albert Einstein"@de ;
  dbo:recievedAward  fbase:m.0jcx__24 ;
  owl:sameAs         dbr:Albert_Einstein , dbr-fr:
      Albert_Einstein , dbr-de:Albert_Einstein ;
  skos:altLabel      "A. Einstein"@fr , "Einstein, Albert"@de ,
      "Albert Einstin"@en ...
```

Listing 1: Example of a fact in FactBench.

finding, which validates input RDF triples (Listing 1) by finding potential sources for them on the Web. To this end, DeFacto combines three strategies: (1) searching for textual occurrences of parts of the statements, (2) trying to find web pages which contains the statement expressed in natural language as well as (3) searching for similar content based on structured data (using the `sameas` service as backend). Moreover, DeFacto has been designed to exploit the multilingual characteristic of the Web (see Figure Figure 1). The output of the framework consists in an overall confidence score for the input statement and a set of (excerpts of) relevant web pages which allows the user to manually judge the presented evidence. In addition, the temporal scope of the events is taken into account [4], i.e., DeFacto can estimate in which time frame the fact was valid.

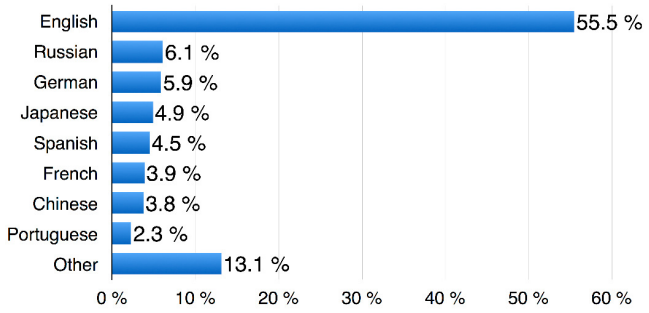


Fig. 1: Usage of content languages for web pages. (W3Techs.com, 21 November 2013)

2 Methodology

The DeFacto core implementation consists of the components depicted in Figure 2. In this example, the system receives an input triple as RDF (*"Nobel Prize was awarded to Albert Einstein"*) and outputs, as **evidence**, a *confidence value* and a *set of (excerpts) web pages* as possible sources for confirmation as well as *meta-information* on the pages. This generated **evidence** enables the user to quickly obtain an overview of possible trustful sources for given statement, instead of having to use search engines, browsing several web pages and looking for relevant pieces of information (all details in [2,3]). Figure 3 depicts the schema for the output provenance information.

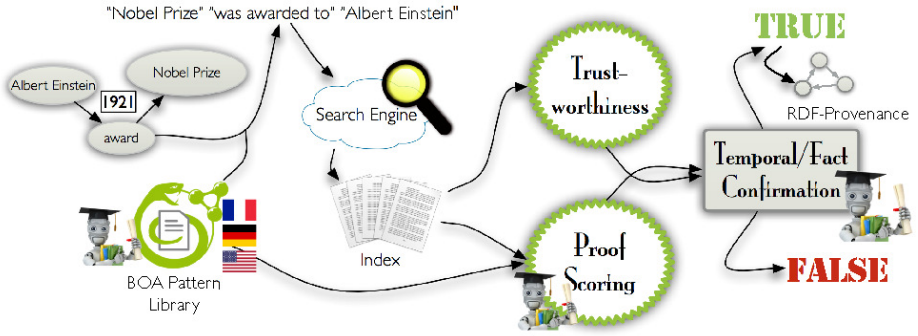


Fig. 2: Overview of the DeFacto's architecture.

3 DeFacto Application

Implemented as a open source[3] single-page application, the application consists of 3 modules, (1) a graphical user interface (GUI), (2) a RESTful Web service (RWS) and

[3] application source code: `http://github.com/AKSW/DeFacto`

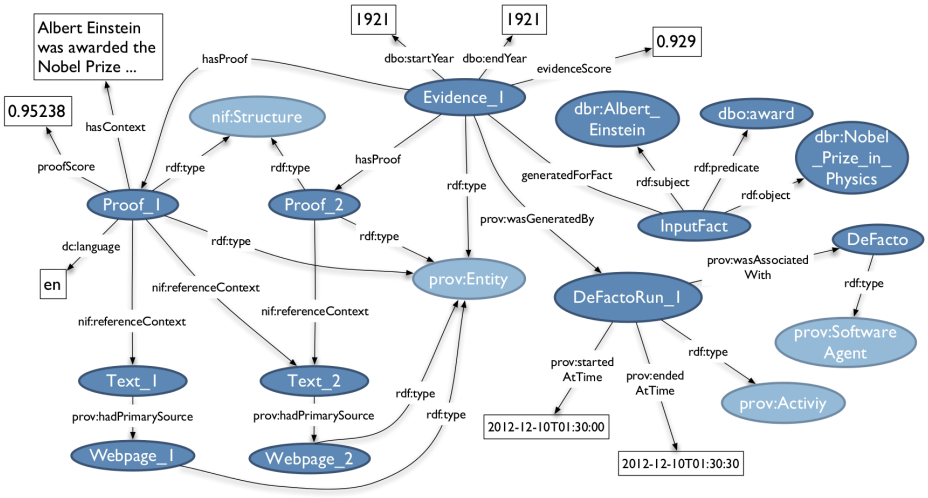
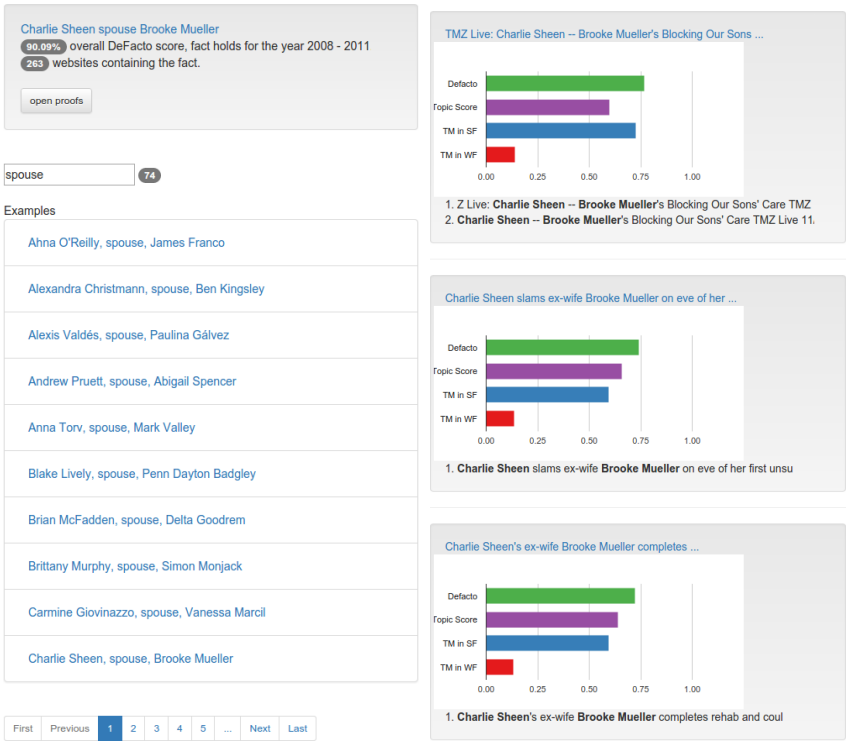


Fig. 3: Overview of the provenance schema which is used to export the validation result of DeFacto as RDF, given the input fact `Albert Einstein, award, Nobel Price in Physics`.

(3) the DeFacto core algorithm. In its current state, the application supports 10 different relations: *award, birth, death, foundation place, leader, nba team, publication date, spouse, starring, subsidiary*. The GUI depicted in Figure 4 allows users to select one of 746 example triples from FactBench, which is then sent to the DeFacto RWS (Figure 4a).

The RWS feeds the DeFacto core algorithm with all the data it needs and starts the core algorithm. The results are transformed to JSON format and are sent back to the GUI to shown to the user. In Figure 4a above the search field, the results are included with the overall DeFacto score, the number of websites that were found and the timeframe within which this fact is assumed to be valid. For more details, a user can also open the proofs by clicking the button below that information. That will open a list of all the websites and their scores, keywords and a link to the website (Figure 4b). For advanced users, a web service which allows sending one's own triple directly to the RWS to request either JSON or RDF results is available. More details about the use of the RWS is provided in the documentation [4]. The service also allows extending DeFacto with searches through local corpora. Figure 5 depicts a component diagram of the implemented DeFacto pipelines.

[4] RESTful Web service documentation: `http://github.com/AKSW/DeFacto/blob/fusion/defacto-restful/doc`



(a) Search form. (b) Result list.

Fig. 4: DeFacto GUI: overall score (a) and proofs for the input fact (b)

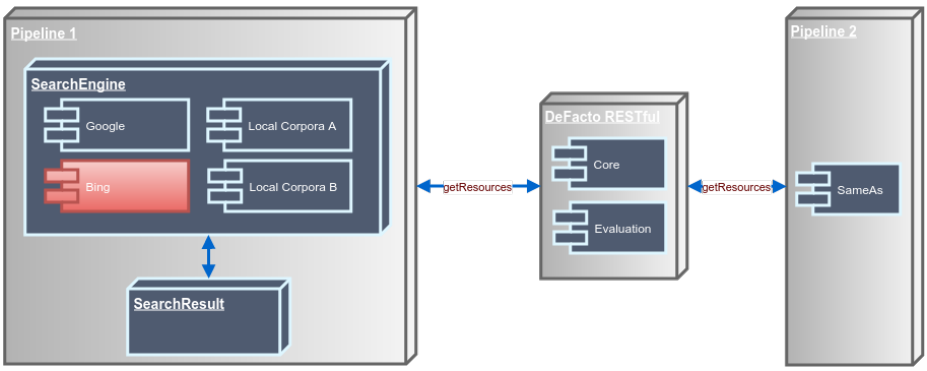


Fig. 5: Searching pipelines for DeFacto: using the `SameAs` service as back-end for obtaining similar resources. The red `SearchEngine` component highlights the used search engine API (Bing) for obtaining the web pages

4 Conclusions and Future Work

We presented a RESTful interface for DeFacto, a multilingual and temporal approach for checking the validity of RDF triples using the Web and Linked Data Resources as background corpus. The approach provides a simple GUI for non-experts as well as a RESTful service for expert users, which allow integrating DeFacto into other applications. The implemented service further allow porting the DeFacto kernel to localized corpora, therewith ensuring scalability and flexibility. Furthermore, the RESTful implementation minimizes the effort level needed for implementing new features and pipelines.

Acknowledgments This work was supported by grants from the European Union's 7th Framework Programme provided for the projects GeoKnow (GA no. 318159), the Eurostars project DIESEL (E!9367) as well as the German Research Foundation Project GOLD and the German Ministry of Economy and Energy project SAKE (GA No. 01MD15006E). We also would like to thank the CAPES foundation, Ministry of Education of Brazil (n: BEX 10179/13-5).

References

1. Amrapali Zaveri, Anisa Rula, Andrea Maurino, Ricardo Pietrobon, Jens Lehmann, Sören Auer, and Pascal Hitzler. Quality assessment methodologies for linked open data. *Semantic Web Journal*, 2015.
2. Jens Lehmann, Daniel Gerber, Mohamed Morsey, and Axel-Cyrille Ngonga Ngomo. Defacto - deep fact validation. In *Proc. of the International Semantic Web Conference*, 2012.
3. Daniel Gerber, Diego Esteves, Jens Lehmann, Lorenz Bühmann, Ricardo Usbeck, Axel-Cyrille Ngonga Ngomo, and René Speck. Defacto—temporal and multilingual deep fact validation. *Web Semantics: Science, Services and Agents on the World Wide Web*, pages –, 2015.
4. Anisa Rula, Matteo Palmonari, Axel-Cyrille Ngonga Ngomo, Daniel Gerber, Jens Lehmann, and Lorenz Bühmann. Hybrid acquisition of temporal scopes for RDF data. In *ESWC*, pages 488–503, 2014.
5. Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N. Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick van Kleef, Sören Auer, and Christian Bizer. DBpedia - a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web Journal*, 6(2):167–195, 2015.
6. Xiaoxin Yin, Jiawei Han, and Philip S. Yu. Truth discovery with multiple conflicting information providers on the web. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1048–1052, 2007.
7. Xin Luna Dong, Laure Berti-Equille, and Divesh Srivastava. Truth discovery and copying detection in a dynamic world. *PVLDB*, 2:562–573, 2009.
8. Alban Galland, Serge Abiteboul, Amélie Marian, and Pierre Senellart. Corroborating information from disagreeing views. In *WSDM*, pages 131–140. ACM, 2010.

Appendix: Addressing Open Track Requirements

The requirements are in the exact order listed on the 2015 challenge criteria web page.

4.1 Minimal Requirements

A. *End User Application and Use of Semantic Technologies:* DeFacto provides a REST Interface for expert-users to integrate fact validation in third party applications. It also provides a website which is suitable for non-expert users. Generally, end users must be able to verify whether text snippets support a particular statement which is usually a strength of humans. DeFacto provides help for understanding the various indicators it shows as well as its input format. Semantic technologies are used in the back-end, e.g. RDF datasets are used as sources themselves, the input is converted to RDF and the BOA pattern library uses RDF.

B. *Information Sources:* The data includes heterogeneous resources gathered from different data sources. In our experiments, Defacto combines data from *DBpedia* [5], the *Web* in general (via search engine queries) and the *Web of Data* (via fact validation through interlinks and similarity).

C. *Semantic Analysis:* DeFacto uses BOA as back-end for semantic enrichment of the data, which allows extracting structured data from the human-readable Web by using Linked Data as background knowledge. Besides, it benefits from the generated natural language patterns for formal relations that allows bridging the gap between structured and unstructured data. DeFacto also uses the sameAs.org service for not only finding evidence in natural language websites, but also in already RDF structured datasets. Three machine learning and training processes are part of DeFacto and form an intelligent backbone (specifically BOA pattern training, proof scoring via machine learning and the overall confidence estimation).

4.2 Additional Desirable Features

- *The application provides an attractive and functional Web interface (for human users):* DeFacto provides a web site which requires the user to be able to phrase its query as a fact (or use one of the many predefined examples) and to give feedback (correct/incorrect). Despite the complexity of the actual algorithms, those limited user interactions are feasible for typical users such as data journalists. We considered it essential to focus on those main features.
- *The application should be scalable (in terms of the amount of data used and in terms of distributed components working together). Ideally, the application should use all data that is currently published on the Semantic Web:* DeFacto has been designed to scale in terms of the number of datasets to be processed and individual dataset size. Its architecture is modular, which allows the generation of different experimental setups. DeFacto currently uses the part of the web which can by typically accessed via search engines as well the part of the data web for which links are stored in the sameAs.org service. Due to this input, DeFacto typically requires 30 to 180 seconds for its analysis.
- *Rigorous evaluations have taken place that demonstrate the benefits of semantic technologies, or validate the results obtained:* The FactBench benchmark is the first available resource that could be used for public fact checking experiment analysis. This benchmarks consists of one training and several test sets for fact validation as well as temporal scope detection. The approach achieves an F_1 measure of 84.9%

on the most realistic fact validation test set (FactBench *mix*) on DBpedia as well as Freebase data. The challenge submission focuses on the interface - a report with all experimental details is available (see [3]).

- *Novelty, in applying semantic technology to a domain or task that have not been considered before:* While fact validation as such is not completely novel (e.g.: *TruthFinder* [6], *AccuVote* [7] and *3-Estimates* [8]), DeFacto is the first framework to perform RDF based fact validation including the usage of structured RDF data in addition to web retrieval.
- *Functionality is different from or goes beyond pure information retrieval:* The combination of NLP functionality, search over structured and unstructured data, three machine learning tasks with a variety of features lets DeFacto go far beyond pure information retrieval. The interface presents the end result of the various analytical steps in the DeFacto pipeline.
  Contextual information is used for ratings or rankings: For proof scoring (i.e. evaluating whether a text excerpt contains a fact), DeFacto uses several context features which are then weighted by various machine learning approaches (out of which decisions trees scored highest).
- *There is a use of dynamic data (e.g. workflows), perhaps in combination with static information:* DeFacto performs search engine queries and Linked Data resource dereferencing as part of its analysis. In this sense, DeFacto adapts to changes in the (data) web dynamically.
- *The results should be as accurate as possible:* DeFacto was carefully trained and achieved high scores on FactBench. However, there is no guarantee of DeFacto performing close to maximum accuracy as RDF fact validation is a rather novel area. Various extensions are possible ranging from explicit negation detection over sentiment analysis, HTML structure detection etc. However, even in its current form DeFacto uses sources across different languages, supports the detection of temporal aspects, uses a multitude of datasets and websites for finding evidence and was trained substantially yielding a close to 90% F-score on FactBench.
- *There is support for multiple languages and accessibility on a range of devices:* DeFacto verbalizes the fact by using multi-lingual natural-language patterns extracted by the BOA Framework. Given a French input fact, it can search in e.g. German and English sources and combine evidence. In terms of devices, DeFacto supports those which can run web browsers.