HAL
open science

# Towards Linked Data Extraction From Tweets

Manel Achichi, Zohra Bellahsene, Dino Ienco, Konstantin Todorov

## To cite this version:

Manel Achichi, Zohra Bellahsene, Dino Ienco, Konstantin Todorov. Towards Linked Data Extraction From Tweets. EGC: Extraction et Gestion des Connaissances, Jan 2015, Luxembourg, Luxembourg. pp.383-388. hal-01411403

**HAL Id: hal-01411403**

**https://hal.science/hal-01411403v1**

Submitted on 7 Dec 2016

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Towards Linked Data Extraction From Tweets

Manel Achichi*, Zohra Bellahsene*
Dino Ienco**, Konstantin Todorov*

*Université Montpellier 2, LIRMM
firstname.lastname@lirmm.fr,
**IRSTEA Montpellier, UMR TETIS
dino.ienco@teledetection.fr

**Abstract.** Millions of Twitter users post messages every day to communicate with other users in real time information about events that occur in their environment. Most of the studies on the content of tweets have focused on the detection of emerging topics. However, to the best of our knowledge, no approach has been proposed to create a knowledge base and enrich it automatically with information coming from tweets. The solution that we propose is composed of four main phases: topic identification, tweets classification, automatic summarization and creation of an RDF triplestore. The proposed approach is implemented in a system covering the entire sequence of processing steps from the collection of tweets written in English language (based on both trusted and crowd sources) to the creation of an RDF dataset anchored in DBpedia's namespace.

## 1 Introduction

One of the goals of the Linked Open Data (LOD) initiative is to structure and interconnect data on the web by using semantic web technologies, such as the Resource Description Framework (RDF), thus taking the web of today up to a new level where data are interpretable and accessible by both humans and machines (Bizer et al., 2009). A considerable effort has been made in that direction throughout the last couple of years. However, many sources of valuable information on the web still remain unexplored, although they contain useful data that can be beneficial for the LOD project. In this paper, we focus on the social medium Twitter that provides a platform for the publication of short messages (*tweets*) of maximal length of 140 characters. The network has enjoyed a growing popularity through the past years, becoming a major source of novel information about many important events, made available in real time, often even before its diffusion through the conventional broadcasting channels. The main motivation of our work is to enable the integration of the information flowing daily through the Twitter stream to the web of data. We propose a method for the extraction of relevant data from Tweets, their conversion into RDF and their storing into an RDF triplestore with the final objective of their publication as linked open data. Our approach covers the entire processing chain following a well-defined workflow, which will be presented in Section 3.

## 2 Related Work

A big family of related approaches focuses on **relation extraction from text**, divided into those based on dependency and those based on syntactic parsing. We can cite systems like Reverb (Fader et al., 2011) or the multilingual DepOE (Gamallo et al., 2012) from the latter group and CLAUSIE (Corro and Gemulla, 2013) – from the former. The system OLLIE (Schmitz et al., 2012) is based on relation patterns which are extracted by Reverb.

There are two main categories of approaches to **extract RDF triples from text**. The first category exploits background knowledge to infer new facts from text. One example is the method proposed in (Anantharangachar et al., 2013), where RDF triples are extracted using domain specific dictionaries induced by an existing ontology. The methods belonging to the second category usually apply semantic analysis on text (Exner and Nugues, 2012; Augenstein et al., 2012; Cattoni et al., 2012). In (Cattoni et al., 2012), a large-scale infrastructure to store and interlink multimedia resources is presented. The system is able to import and annotate knowledge in the form of RDF, automatically associating resources to entities and creating new knowledge in the form of RDF triples. One particularity of this system is the association of context information to each managed resource.

We distinguish between two approaches to **tweets summarization** – the first group provides a bag of terms as a summary (Cataldi et al., 2013; Benhardus and Kalita, 2013; Mathioudakis and Koudas, 2010), while the second extracts representative tweets. Approaches of the second group are more suitable to us, as they preserve some kind of structural coherence and conciseness in the derived summary. In (Sharifi et al., 2010), a set of tweets is summarized by a sentence derived by a graph representation of the words (co-)occuring in a collection of tweets. (Chua and Asur, 2013) retrieve the most relevant tweets as summary of a tweets collection, based on topic models. Two other techniques are proposed in (Olariu, 2013). The first one first merges all the tweets in a word graph (similarly to (Chua and Asur, 2013)) and then computes a score function to select a path of the graph as a possible summary. The second one selects the most frequent words subsequence (phrase).

## 3 Overview of the Approach

Our processing chain covers the entire process from the constitution of a Twitter corpus to the generation of an RDF triplestore (Figure 1).

**Constituting a corpus of tweets.** Tweets written in English are first collected from *trusted sources* – Twitter accounts of established media (e.g., the BBC). These tweets are then grouped together topic-wise, by using topic identification techniques, forming homogeneous clusters. We have chosen the K-MEANS clustering algorithm for its simplicity and efficiency. We set K at the lowest value that maintains the ratio {average intra-cluster distance / average inter-cluster distance} under a given threshold. Further, for every topic, we collect *crowded tweets* from ordinary users accounts, which are used to enrich the information contained in every topic. Finally, we only keep the tweets closely related to topics that we want to represent, i.e., tweets containing previously extracted keywords (the most frequent words in each cluster).

**Filtering and preprocessing.** A filtering component is applied to both trusted and crowd sources. The aim is to only keep tweets that contain named entities, and particularly ones that have DBpedia URIs, in preparation of the linking of the (yet to be) extracted RDF triples to

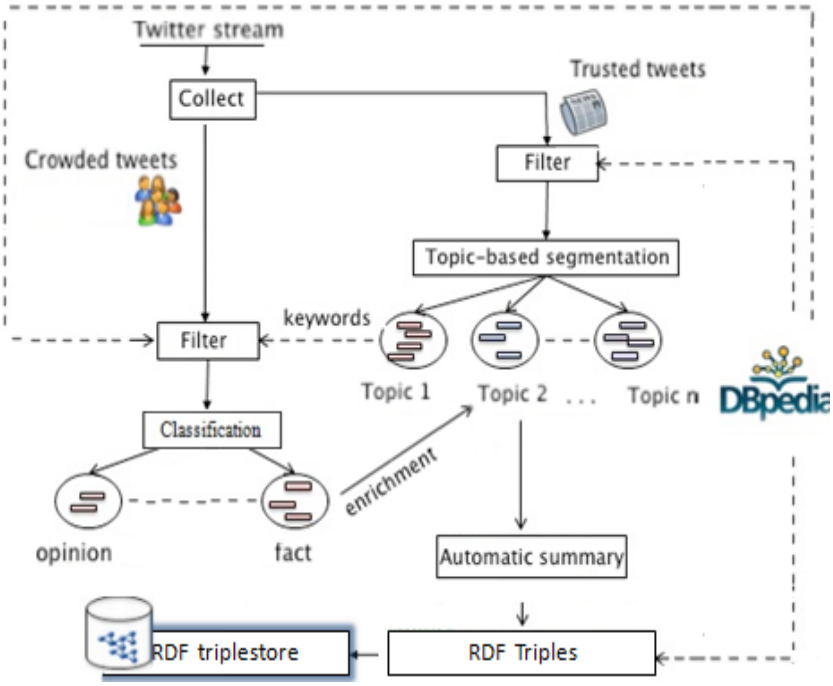


FIG. 1 – *The workflow of the system.*

the web of data. The crowded tweets are additionally filtered with respect to their adequacy to a given topic by the help of a set of keywords describing each topic. Moreover, all tweets are preprocessed by removing hashtags, urls and retweets and by lemmatizing the text.

**Crowded tweets classification.** We introduced a text classification module, based on the principle of sentiment analysis, in order to classify the crowded tweets into two categories: (i) *fact*, including neutral, objective information that we want to keep and use to enrich the trusted tweets, and (ii) *other*, including opinion, private messages, etc. We used the *coreNLP* tool. Here, we assume that a tweet that conveys factual information (or a piece of news) is stripped from both positive and negative sentiment. However, assessing the polarity of a tweet was not sufficient in order to judge on its objectivity. As additional classification criteria, we check for the absence of the following set of features within each tweet: (i) the "@" symbol indicating a private message, (ii) repetitive letters or abuse of ponctuation, (iii) smileys that reflect a mindstate, and (iv) misspelled words.

**Automatic summarization.** After constituing a tweets corpus per topic, we proceed to generate a summary of this corpus, in order to remove redundant and unnecessary information. The algorithm that we have developed generates automatically a summary of a topic in the form of a concise and coherent set of tweets. It takes two parameters as input: a set of tweets corresponding to a topic and a threshold to control tweets similarity ($\sigma$). First, the algorithm constructs a weighted undirected graph ($G$) where the vertices are the tweets and an edge exists between two vertices if the cosine similarity between the two corresponding tweets is

greater than $\sigma$. Further, the algorithm proceeds to extract the maximal cliques from $G$, relying on the well-known *Bron-Kerbosch* algorithm. The assumption here is that a maximal clique represents a cluster of potentially redundant information that can be represented by one single tweet. The PageRank algorithm is then used to assign scores to the vertices within each clique and select the one with the highest score as a summary of the clique. If we have several tweets with the same score, the algorithm selects the longest one. The summary of the entire topic is given by the set of tweets representing each maximal clique within the topic graph.

**From tweets to RDF.** Finally, the last step consists in transforming every summary into an RDF graph. We were inspired by the state-of-the-art approach to RDF extraction from text, LODifier (Augenstein et al., 2012), since it relies on the idea of anchoring the extracted semantic information in DBpedia's namespace. Our approach consists of four main steps:

**(1) Semantic analysis.** The semantics of every phrase is modeled as a (set of) triple(s) of the kind $\langle$*subject, verb, object*$\rangle$. The algorithm given in (Rusu et al., 2007) was adopted and adapted because of its effectiveness combined with the Stanford Parser. This algorithm works well with simple single-clause sentences but fails to handle properly multi-clause sentences. We propose the following clause-splitting iterative algorithm : (1) Apply dependency analysis on the complex phrase. (2) For each dependency relationship of type "nsubj" (nominal subject), retrieve all relationships of its arguments, except for the "nsubj" type relationships. (3) Repeat (2) for each retrieved relationship until no new relationship is added. (4) For all dependencies enveloped by a relationship of type "nsubj", extract and organize all the words in ascending order of the numbers associated to them. These numbers indicate the order of words in the original sentence.

**(2) Disambiguation.** Before assigning a DBpedia URI to a word, we choose its most appropriate sense in a given context. We have applied a commonly used method based on synset identification in WordNet.

**(3) DBpedia URI assignment.** We assign to every term its corresponding DBpedia URI.

**(4) Generation of an RDF graph.** During the semantic analysis, the sentences were decomposed to simple clauses and each clause has been structured as a triple of the kind $\langle$*subject, verb, object*$\rangle$. In this last step, the system performs a conversion of these triples in an RDF graph. Arguments created during the semantic analysis are analyzed to identify the named entities that correspond to RDF subjects and objects and assign to each of them the corresponding DBpedia URI. To do this, we match the subject and the object of the extracted triples to the existing triples in the DBpedia graph.

# 4 Prototyping and Experiments on Real Twitter Data

We use the Twitter4J [1] API to collect tweets from stream, the TextRazor [2] service to recognize entities and successively assign them a Wikipedia URL, and the Stanford CoreNLP library to deal with natural language processing and sentiment analysis.

The collection of *trusted tweets* is obtained by crawling the social medium on October 9, 2014 for a period of 24 hours. We followed the accounts of BBC World, CNN, New York Times, New York Times World, and Breaking News.We kept those tweets that contain at least one named entity that corresponds to a DBpedia URI. This first collection is composed by

1. `http://twitter4j.org/en/index.html`
2. `https://www.textrazor.com/`

125 tweets. The collection of *crowded tweets* is retrieved on October 10, 2014 by considering keywords extracted from the *trusted tweets*. In our dataset, we were able to detect 50 different topics from the trusted tweets. Here is an example of a topic composed by 5 tweets: {*(1) Swimming made Michael Phelps a dominant athlete but it couldn't guide him outside of the pool; (2) USA Swimming announces 6 months suspension for Michael Phelps after DUI arrest; (3) Michael Phelps suspended by USA Swimming; (4) Michael Phelps received a six months suspension from swimming; (5) Swimmer Michael Phelps suspended for six months.*}. For that example, we retrieved 1 325 short messages from crowded sources by using trusted tweets keywords. Successively, we classify the tweets as *news* or *other* by applying the classification rules presented in Section 3. Table 1 reports the results of the classification. We obtain an accuracy of $76.22\%$. We also analyze the behavior of our method by only considering the class *news*. To this end, we compute *Precision*, *Recall* and *F-Measure* for this class obtaining, respectively, $0.628$, $0.722$ and $0.672$. This evaluation underlines the quality of our strategy and shows that our method is able to detect about 2 out of 3 tweets containing factual information.

| | | Predicted Class | |
|---|---|---|---|
| | | **News** | **Other** |
| Real Class | **News** | 323 | 124 |
| | **Other** | 191 | 687 |

TAB. 1 – *Confusion matrix obtained by our classification strategy on the collected tweets.*

| **Subject** | **Predicate** | **Object** |
|---|---|---|
| Michael Phelps | be | athlete |
| Michael Phelps | receive | suspension |
| Michael Phelps | arrest | Baltimore |
| Michael Phelps | arrest | drink |
| Michael Phelps | take | break |
| Michael Phelps | suspend | swimming |

TAB. 2 – *Examples of RDF triples automatically extracted by our framework.*

We note that low values (between 0.3 and 0.5) of the $\sigma$ parameter correspond to a high number of maximal cliques, while the trend for values between 0.5 and 0.95 is quite stable.

In Table 2, we show several triples produced by our method with $\sigma$ equaling 0.65. These triples represent information that is automatically induced by the set of tweets on the topic *Michael Phelps*. In total, 10 triplets were extracted of which 4 triplets are identical. The small number of extracted triples is due to the fact that collecting the tweets was performed during 24 hours only. All the RDF triples are then stored in a triple store [3], ready to be published and interlinked on the web of data.

# 5 Conclusion

The approach that we propose is complete – we take as an input the heterogeneous stream of tweets flowing daily through the social medium and we output an RDF triplestore containing only factual information about entities that live in DBpedia's namespace. Among the original contributions of our framework, we underline: i) the use of two main sources of data – trusted (tweets coming from established mainstream media) and crowd (ordinary users tweets). The latter source is used to enrich the data collected from the former. In order to limit redundancy

3. Virtuoso, `http://virtuoso.openlinksw.com/`

in the collected information and prepare the corpus of tweets for the RDF triples extraction phase, ii) we introduce a novel approach to automatically generate a summary of a set of tweets, relevant to a given topic. As the extracted summaries can be rich and structurally complex, iii) we design a strategy to split multi-clause tweets into simple chunks of information in order to easily identify the components of a RDF triple. Finally, the entire approach is implemented in a modular prototype.

## References

Anantharangachar, R., S. Ramani, and S. Rajagopalan (2013). Ontology guided information extraction from unstructured text. *IJVeST 4*(1), 19.

Augenstein, I., S. Padó, and S. Rudolph (2012). Lodifier: Generating linked data from unstructured text. In *ESWC*, pp. 210–224.

Benhardus, J. and J. Kalita (2013). Streaming trend detection in twitter. *IJWBC 9*(1), 122–139.

Bizer, C., T. Heath, and T. Berners-Lee (2009). Linked data - the story so far. *IJSWIS 5*(3), 1–22.

Cataldi, M., L. D. Caro, and C. Schifanella (2013). Personalized emerging topic detection based on a term aging model. *ACM TIST 5*(1), 7.

Cattoni, R., F. Corcoglioniti, C. Girardi, B. Magnini, L. Serafini, and R. Zanoli (2012). The knowledgestore: an entity-based storage system. In *LREC*, pp. 3639–3646.

Chua, F. C. T. and S. Asur (2013). Automatic summarization of events from social media. In *ICWSM*.

Corro, L. D. and R. Gemulla (2013). Clausie: clause-based open information extraction. In *WWW*, pp. 355–366.

Exner, P. and P. Nugues (2012). Entity extraction: From unstructured text to dbpedia rdf triples. In *WoLE*.

Fader, A., S. Soderland, and O. Etzioni (2011). Identifying relations for open information extraction. In *EMNLP*, pp. 1535–1545.

Gamallo, P., M. Garcia, and S. Fernández-Lanza (2012). Dependency-based open information extraction. In *Workshop on Unsupervised and Semi-Supervised Learning in NLP*, pp. 10–18. Association for Computational Linguistics.

Mathioudakis, M. and N. Koudas (2010). Twittermonitor: trend detection over the twitter stream. In *SIGMOD*, pp. 1155–1158.

Olariu, A. (2013). Hierarchical clustering in improving microblog stream summarization. In *CICling*, pp. 424–435.

Rusu, D., L. Dali, B. Fortuna, M. Grobelnik, and D. Mladenic (2007). Triplet extraction from sentences. In *Int. Multiconf. Information Society-IS*, pp. 8–12.

Schmitz, M., R. Bart, S. Soderland, O. Etzioni, et al. (2012). Open language learning for information extraction. In *EMNLP*, pp. 523–534.

Sharifi, B., M. Hutton, and J. K. Kalita (2010). Summarizing microblogs automatically. In *HLT-NAACL*, pp. 685–688.