Generating RDF Metadata from Twitter Streams

Teklu Aregawi Gidey[1], Toshiyuki AMAGASA[2]
1 Graduate School of Science and Technology
2 Center for Computational Sciences
University of Tsukuba

Abstract

In recent years, Social Network Service has gained much public attention as a source of information. Twitter is considered as most prominent SNS source. Most of previous studies on the content of tweets have focused on the detection of emerging topics and some have focused on creation of RDF. It is important to generate RDF metadata from tweets to put data in context via linking and semantic metadata providing a framework for data integration, analytics and sharing. However, in the previous studies, the generated RDFs were not from streaming tweets and the mentions interlinked with knowledge graphs did not show the location of the text in the tweet. In our experiment, we generated RDF metadata from streaming tweets using a hashtag 'Covid-19' and we located the exact position of the mentions in the tweet being linked to a knowledge graph, WikiPedia using TagMe entity linker. We expressed it as a URI which includes the URL of Twitter, the user name, the tweet id and index of the offset information.

1 Introduction

We are living in the era of technology, where sources of information are given a priority to get timely data on day to day activities of the globe. Social Network Service (SNS) is one which is gaining much public attention and being regarded as an important source of information. SNSs are emerged as an essential forums for negotiating and giving comments about news, events happening around the world. Such user generated information can be important data source for researches in different fields such as data science, sociology, psychology and historical studies that can be used by researchers to understand behavior, trends and opinions [1].

Twitter is considered as most prominent SNS source. Twitter is a microblogging which is growing rapidly in the last years which is considered as online social network service that counts about 336 million monthly active users at the beginning of 2018 [2]. Millions of Twitter users post messages every day to communicate with other users in real time information about events that occur in their environment [8]. Due to the fast evolution of twitter, researchers study tweets' content characteristics, so this can help in extraction of information like users' topics, opinions about topics [3][4]. Entities like user Hashtags, mentions, URLs associated with a tweet, places represent the locations in the real word, so tweets play essential role in twitter data analysis [9].

Most of previous studies on the content of tweets have focused on the detection of emerging topics and some have focused on creation of RDF Knowledge Graphs [8]. Now a days, Knowledge Graph is becoming popular and important topic in many domains. A knowledge graph represents a collection of interlinked descriptions of entities – objects, events or concepts. Knowledge graphs put data in context via linking and semantic metadata and this way provide a framework for data integration, unification, analytics and sharing [10]. Knowledge graphs are represented in RDF (Resource Description Framework). 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 [11]. RDF is based on the idea of making statements about resources (web resources) in expressions of the form subject–predicate–object, known as triples. Figure 1 shows the general form of RDF representation in triples, where subject refers to the resource identifier, predicate refers to the property of that resource and an object refers to the value of that property which belongs to the resource. For example, in the statement

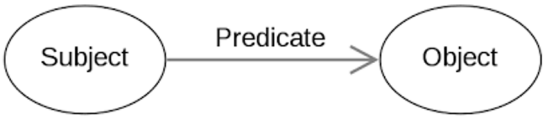

Figure 1: RDF representation in Triples form

"The sky has the color blue", sky is a subject, has color is a predicate and blue is an object.

It is important to generate RDF metadata from SNSs like Twitter. However, to the best of our knowledge, the generated RDFs, in the previous studies, were not from streaming tweets and the mentions interlinked with knowledge graphs did not show the exact location of the text in the tweet. Streaming tweets help us to get updated and timely information on events or news based on hashtags. We then proposed a way to generate RDF metadata from streaming tweets.

Contributions of our work include:
- Generating RDF metadata from streaming tweets
- Locating the position of a mention in a text of the tweet in a URI form, expressed by the combination of the URL of twitter, user name of the one who tweeted it, tweet id and start and end indexes of the text location.

The remaining part of this paper is organized as follows: Section 2 describes the related works in the area. Section 3 outlines the proposed approach for generating RDF metadata and the required steps for developing a knowledge graph of interlinked events. Section 4 frames the implementation details with some running examples. Section 5 concludes the paper with suggestions for future work.

2 Related Works

In most of previous studies, researchers have focused on the detection of emerging topics on the contents of tweets. Later on, many researchers started proposing different approaches to create a knowledge base and enrich it automatically with information coming from tweets. This has studied in [8], which is focused in topic identification, tweets classification, automatic summarization and creation of an RDF triple store. In this work, although it generates RDF triples and RDF graph from the given tweets, it is not streaming. The work in [12] presents principles for developing a knowledge graph of interlinked events using the case study of news headlines published on Twitter which is a real-time and eventful source of fresh information. This work focuses on choosing background data model, event annotation (i.e., event recognition and classification), entity annotation and eventually interlinking events. The international journal in [7] presents a procedure how to represent twitter information using RDF. How ever, the tweets are not of streaming tweets and it did not interlink with knowledge graphs like our work. The work in [15] links named entities on Twitter to knowledge graphs, but unlike our paper it does not focus on RDF generation, it is not from streaming tweets and it did not show the exact position of mentions. Other than from Twitter, other works have been also proposed which mainly focus on how to create RDF triples and RDF knowledge graphs from texts or sentences and we assume they are helpful works to our work as they explain methods how to generate RDF. SDM-RDFizer [14], RML [16] and RocketRML [17] are RDF mapping languages to create RDF knowledge graphs from heterogeneous data. However, it is not based on streaming. RDF-Gen [19] also generates RDF triples from streaming and archival data. How ever, it did not include the interlinking with a knowledge graph. Some other works focuses on extracting triples from sentences [5] and some on extracting triples from texts [6]. They all work by importing a file or copying and pasting texts as an input and generate an RDF file. They also are not streaming and not able to refer to mentions or to show the exact position of the text in the tweet being linked to knowledge graph. To overcome such gaps, we propose a method to generate RDF metadata from twitter streams which is real-time, generate RDF triples from the streaming tweets and link to a knowledge graph, Wikipedia.

3 The Proposed Approach

3.1 Overview

The proposed work uses streaming tweets as an input and it generates RDF metadata. It also extract, annotate and mention entities and link them with a knowledge graph, Wikipedia. The mentioned text interlinked with the knowledge graph is then represented using URI which includes Twitter URL, user name of the owner of the tweet, tweet ID and index showing the location of the mentioned text. The overall process has two parts: One part explaining a way to generate RDF triples from the streaming tweet. As explained in [7], this part shows a step by step process to generate an RDF metadata. The second part explains a way to interlink with a knowledge graph; Wikipedia, DBPedia, Wikidata or others (in our case, a Wikipedia) and express it using a URI to show the exact location of the mention back in the original tweet. Figure 2 shows the general overview of RDF data generation and Figure 3 shows the general overview of the interlinking.

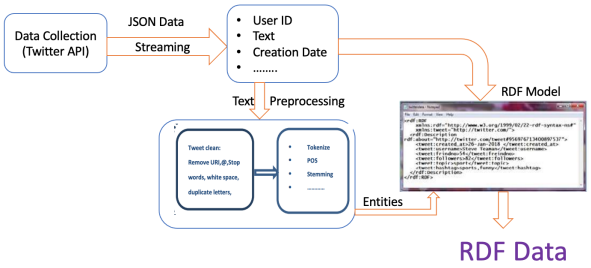

Figure 2: Overview of generating RDF data [7]

3.2 Data Collection

We have used Twitter streaming API, to collect streaming twitter JSON data. We have also used a Python library called Tweepy to access streaming API to download a real-time stream of Tweets. A physical storage is not needed to store the data, it is processed in real time. This streaming JSON data contains many details such as creation date, user id, str id, text (tweet content) and other additional information. All these details may not be needed for preprocessing, so it is required to parse

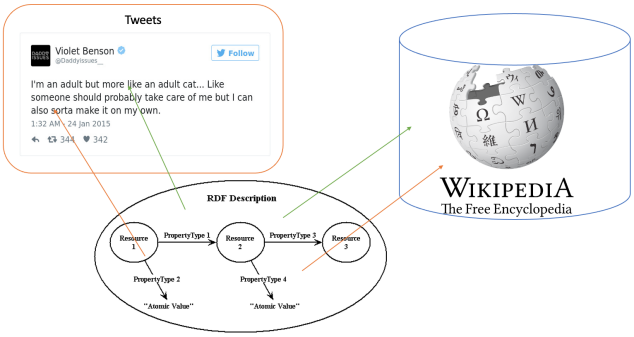

Figure 3: General overview of Linking mentions to a knowledge graph, Wikipedia

the tweets. We used **tweet_parser** package to parse the tweets to extract specific information from tweet object like user name, text, Hashtags, etc. The parsing is done only for English tweets.

3.3 Tweet Text Preprocessing

The main objective of this step is to use the natural language processing techniques to process the input tweet text and make it suitable for the next step to extract the entities correctly. It includes four sub-steps:

Step 1 - Tokenization: it is a process of splitting the input text into separate terms or words, called tokens. Each token can represent word, abbreviation, hyperlink, emoticon or other punctuation symbols that could be found commonly in tweets. We have used NLTK (Natural Language Toolkit) library, one of the best library for preprocessing text data, to tokenize the text data in the tweets. SpaCy library can also be used as an option for preprocessing text data.

Step 2 - Text Cleaning: we have removed any irrelevant textual data from the tweet content itself such as removing repetition in characters, white spaces, URI, symbols like RT,*, &, symbols like semicolon (;), column (,), dots (.), quotations (‘/”), arcs ([, {,},]), numbers, stop words (to, our, and, etc), etc.

Step 3 - PoS Tagging: In this step we extracted the part of speech tags for the input text.

Step 4 - Stemming: we stemmed the words to their original root in order to reduce the initial set of words representing the input tweet text. It involves removing all

suffixes from words (ing, s, tion, etc.) and transferring past tenses to a root word.

3.4 Entity Annotation

Entity annotation is a significant task for creating a knowledge graph of events [12]. We have used TagMe to annotate an entity, as according to [18], TagMe annotate texts which are short and poorly composed, such as snippets of search-engine results, tweets, news, etc and the annotation is extremely informative. An annotation is a pair (spot,entity), where "spot" is a substring of the input text and "entity" is a reference to a Wikipedia page representing the meaning of that spot, in that context. TagMe associates an attribute to each annotation, called ρ (rho), which estimates the "goodness" of the annotation with respect to the other entities of the input text [13]. We stress here that ρ does not indicate the relevance of the entity in the input text, but is rather a confidence score assigned by TagMe to that annotation. You can use the ρ value to discard annotations that are below a given threshold. The threshold should be chosen in the interval [0,1]. A reasonable threshold is between 0.1 and 0.3 [13]. In our experiment, we have used 0.1.

3.4.1 Entity Recognition

Entity extraction, or named entity recognition (NER), is finding mentions of key "things" (aka "entities") such as people, places, organizations, dates, and time within text in the tweet. We can use NLTK or SpaCy to extract entites from the tweet. In our experiment, we have used the SpaCy library. Figure 4 shows an example of how entities can be extracted and represented in a standard way.

Figure 4: Extracted Entities from a given tweet or text

3.4.2 Entity Linking

Entity Linking is about associating entity mentions in a given text to their appropriate corresponding entities in a given knowledge graph [12]. We link enities to knowledge graphs to remove ambiguity. We have used TagMe to associate the mentions to Wikipedia knowledge graph. Each mention is weighted using a factor, called link probability, that measures the reliability of that substring as a significant mention, and this value can be used to refine the returned mentions via a post-processing phase [13]. We have excluded those with very low link probability assuming they are valueless entities and we have considered a link probability value above 0.01.

3.5 RDF Data Generation

3.5.1 RDF Triple Generation

This section is aimed to generate RDF triples from the streaming text in the tweet. It expresses a text in a subject, predicate and object form. There are different tools and libraries to perform this generation, such as RDF-Triple-API, Stanford OpenIE, Stanford dependency parser and others. The text in the tweet is first considered as a group of sentences. The RDF triples are then generated from the sentences. The implementation of this section is not completed yet. We will use Stanford OpenIE [20], a Python library, to generate the RDF triples from sentences found in the text/tweet. As explained in [20], Open information extraction (open IE) refers to the extraction of structured relation triples from plain text, such that the schema for these relations does not need to be specified in advance. For example, Barack Obama was born in Hawaii would create a triple (Barack Obama; was born in; Hawaii), corresponding to the open domain relation "was born in". After installing the library the expected out put for the text 'Barack Obama was born in Hawaii' will be:
$\| - \{$*'subject': 'Barack Obama', 'relation': 'was', 'object': 'born'*$\}$
$\| - \{$*'subject': 'Barack Obama', 'relation': 'was born in', 'object': 'Hawaii'*$\}$
and the expected generated RDF graph is given in Figure 5. Once we finished the implementation for this part, we expect such results, triples in streaming, for the texts in the tweet.

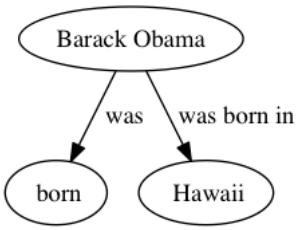

Figure 5: Expected generated RDF graph

3.5.2 RDF Metadata Generation

In the streaming tweet we generate RDF data that comprise different types of metadata such as metadata about the tweet, metadata about attachments and metadata about the texts in the tweet.

The metadata about tweet contains user information, location information and time information. These all can be extracted from the tweet. We generated URI using Twitter URL, user name of the tweet owner and tweet ID from the information we get from the tweet metadata. For example the URI have the form `http://twitter.com/username/status/tweetid`. We used screen name to fetch the user name, tweet id and other details so that it will help us to identify who tweets which tweet during the streaming.

The metadata about the attachments contain the type of the attachment or multimedia, image, audio, video, or link and we encode the reference to the attachments. We can extract Exif information from an image or we can extract object information represented in the image by applying object identification. The Exchangeable image file format (Exif) is a standard that defines the formats of image, audio, and metadata tags used by cameras, phones, and other digital recording devices.

The metadata about the text mainly refers to the entities mentioned in the text as we explained above in the entity linking section. We encoded the mention along its exact position or location in the tweet so that we can go back to the original tweet from the linked knowledge graph. In the entity linking section above we get offset information of the entities, the mentions, using TagMe, we used that offset information to encode the text and added the location of the mention in the tweet. We then added this result to the URI we got from the metadata about tweet above, and give us the form `http://twitter.com/username/status/tweetid#offset_indexBegin_indexEnd`, where offset represents the offset information of the mention from the tweet and indexBegin_indexEnd indicates the position of the mentioned text in the tweet. For example, the position of Mona Lisa in the example above is between 4th and 12th index and hence the URI representation will be `http://twitter.com/username/status/tweetid#offset_4_12`.

The generated RDF triples and the mentions represented using URI are then linked to a knowledge graph, Wikipedia, so that it will have meaning. We have used TagMe entity linker for the interlinking method. The entity linker assigns a unique identity to entities (such as famous individuals, locations, or companies) mentioned in text. Encoding a sample tweet to RDF knowledge graph and interlinking it to a Wikipedia can be shown in Figure 6.

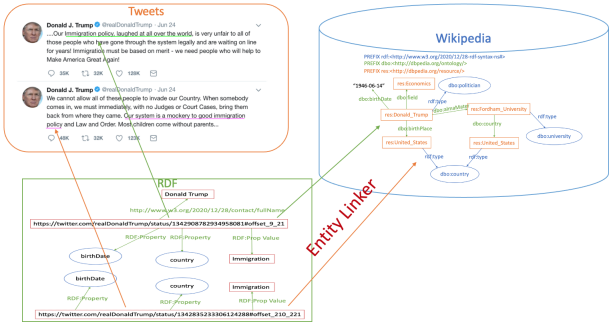

Figure 6: How RDF metadata are generated from stream tweets and interlinked to Wikipedia. The example shows the position of the text ‘immigration’. We use tweet id to distinguish the text ‘immigration’ located in both tweets.

4 Implementation

4.1 Implementation Details

We have used Python 3.7.4 programming language for implementation, PyCharm 2019 2.6 text editor for writing and running the code, Twitter streaming API for fetching streaming tweets, Tweepy to access the API to download

real-time stream of tweets, NLTK library for text preprocessing, SpaCy library for entity extraction, TagMe entity linker for linking mentions to Wikipedia knowledge graph and Stanford OpenIE library for generating RDF triples. Java is also used by the CoreNLP library, where the OpenIE library is available at. In the experiment, we have used Covid-19 as a filtering hashtag text during the streaming data fetch using the Twitter API.

4.2 Running Example

When we run our code, it gives us streaming tweet text about Covid-19, streaming entity extraction from the tweet, streaming mention linking and the URI of the mentions with the exact location of the mention back in the tweet.

Example 1: using the twitter API we fetched a tweet: 'NEW: Number of Americans fully vaccinated against COVID-19 reaches 43%'

From this tweet, by identifying the screen name of the the tweet we can know the user name of the owner who tweeted this and the tweet id and we got user name = Kveldulf7 and tweet id = 1403917288923877376, then the entities extrtacted using SpaCy with their Entity type, the annotated entity with their score and the mentions with their link probability and the URL representing the linking of mentions to Wikipedia with the index showing the exact location of the mentions back in the original tweet is given in Figure 7.

```
Entities using SpaCy
Americans - NORP - Nationalities or religious or political groups
COVID-19 - PRODUCT - Objects, vehicles, foods, etc. (not services)
43% - PERCENT - Percentage, including "%"

Annotations using TagMe
Number -> Continent (score: 0.10470223426818848)
vaccinated -> Vaccination (score: 0.14910714328289032)

Mentions
Americans lp=0.0251001063734293
https://twitter.com/Kveldulf7/status/1403917288923877376#offset_15_24
vaccinated lp=0.09045226126909256
https://twitter.com/Kveldulf7/status/1403917288923877376#offset_31_41
```

Figure 7: Running Example 1

Example 2: another fetched tweet in the streaming is also: 'Participated in the @G7 Summit session on Health. Thanked partners for the support during the recent COVID-19 wave. India supports global action to prevent future pandemics. "One Earth, One Health" is our message to humanity. #G7UK `https://t.co/B4qLmxLIM7`'

From this tweet, user name = KirenRijiju and tweet id = 1403917301846478848, then the result same as in example 1 is shown in Figure 8.

```
Entities using SpaCy
Health - ORG - Companies, agencies, institutions, etc.
COVID-19 - PERSON - People, including fictional
India - GPE - Countries, cities, states
One - CARDINAL - Numerals that do not fall under another type
Earth - LOC - Non-GPE locations, mountain ranges, bodies of water

Annotations using TagMe
G7 -> Group of Seven (G7) (score: 0.37679028511047363)
G7 Summit -> Group of Eight (G8) (score: 0.3143596649169922)
Health -> Health (score: 0.11716138571500778)
wave -> Wave (score: 0.12434493750333786)
India -> India (score: 0.3361321985721588)
global -> Globalization (score: 0.10352487117052078)
action -> Action (fiction) (score: 0.10173925757408142)
future -> Future (score: 0.1882350593805313)
pandemics -> Pandemic (score: 0.25429001450538635)
Earth, One -> Earth-One (score: 0.11205074191093445)
message -> The Message (Animorphs) (score: 0.14240601658821106)
humanity -> Human (score: 0.144758403301239)
https -> HTTPS (score: 0.39057061076164246)
t.co -> Twitter (score: 0.4011480212211609)

Mentions
G7 lp=0.3310810923576355
https://twitter.com/LokeshT24919290/status/1403917306871250944#offset_21_23
G7 Summit lp=0.08275862038135529
https://twitter.com/LokeshT24919290/status/1403917306871250944#offset_21_30
wave lp=0.023498859256505966
https://twitter.com/LokeshT24919290/status/1403917306871250944#offset_110_114
India lp=0.6044615507125854
https://twitter.com/LokeshT24919290/status/1403917306871250944#offset_118_123
action lp=0.01932186633348465
https://twitter.com/LokeshT24919290/status/1403917306871250944#offset_140_146
pandemics lp=0.13656388223171234
https://twitter.com/LokeshT24919290/status/1403917306871250944#offset_165_174
Earth, One lp=0.2241014838218689
https://twitter.com/LokeshT24919290/status/1403917306871250944#offset_183_193
One Health lp=0.10240963846445084
https://twitter.com/LokeshT24919290/status/1403917306871250944#offset_190_200
message lp=0.0120285889133811
https://twitter.com/LokeshT24919290/status/1403917306871250944#offset_209_216
humanity lp=0.016733339056372643
https://twitter.com/LokeshT24919290/status/1403917306871250944#offset_220_228
https lp=0.6907216310501099
https://twitter.com/LokeshT24919290/status/1403917306871250944#offset_236_241
t.co lp=0.6666666865348816
https://twitter.com/LokeshT24919290/status/1403917306871250944#offset_244_248
```

Figure 8: Running Example 2

5 Summary and Future Work

The main advantage of representation of tweets information in RDF format is making semantic meaning of tweets

information to interlink and access all that we know and assert (think data, records, information, facts) in a formal, machine-processable language. The RDF metadata will be generated from twitter streams, will able to locate the exact position of text in the tweet and linked to Wikipedia using a TagMe Entity Linker.

As a future work, we will finish the RDF triple generation and we will evaluate if the results are correct, valid and efficient. For example, Covid-19 is sometimes represented as PERSON, sometimes as PRODUCT, sometimes omitted and represented as others during the entity extraction process as it is a new biological term and not added as new entity type yet. We will try to add Covid-19 as a new entity type by training NER with custom training data. During the streaming, there is also a second or 2 second delay for generating the mentions and their URIs, we will try to find a way to avoid or minimize delays. We know that tens of thousands of tweets are tweeted per second. We will then look for ways to increase efficiency and speed of the process so that it will adapt and work efficiently for the large number of tweets tweeted per second.

References

[1] Fafalios P., Iosifidis V., Ntoutsi E., Dietze S. (2018) TweetsKB: A Public and Large-Scale RDF Corpus of Annotated Tweets. In: Gangemi A. et al. (eds) The Semantic Web. ESWC 2018. Lecture Notes in Computer Science, vol 10843. Springer, Cham. https://doi.org/10.1007/978-3-319-93417-4_12

[2] Statista – The portal for statistics, https://www.statista.com/statistics/282087/number-of-monthly-active-twitter-users/, [Accessed 28 Dec. 2020].

[3] NEBHI, Kamel. Ontology-Based Information Extraction from Twitter. In: Proceedings of the Workshop on Information Extraction and Entity Analytics on Social Media Data - COLING 2012. Mumbai (India). [s.l.] : The COLING 2012 Organizing Committee, 2012. p. 17-22. https://archive-ouverte.unige.ch/unige:24683

[4] Asfari, O., Hannachi, L., Bentayeb, F., Boussaid, O. (2013). Ontological Topic Modeling to Extract Twitter Users' Topics of Interest, ICITA , 141—146.

[5] Rusu, Delia, Lorand Dali, B. Fortuna, M. Grobelnik and D. Mladení. "Triplet Extraction from Sentences." (2007), SiKDD

[6] Exner, P., Nugues, P., From Unstructured Text to DBpedia RDF Triples, ISWC 2012

[7] Asaad Hadi, Enass Abdulshaheed, Twitter Information Representation using RDF, International Journal of Engineering and Technology 2019

[8] 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.

[9] Russell, M. (2013). Mining the Social Web Data Mining Facebook, Twitter, LinkedIn, Google+, GitHub, and more, 2nd ed. O'Reilly Media, p. 448.

[10] D. Fensel, U. Simsek, K. Angele, E. Huaman, E. Kärle, O. Panasiuk, I. Toma, J. Umbrich, A. Wahler, Knowledge Graphs - Methodology, Tools and Selected Use Cases. Springer 2020

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

[12] S. Shekarpour, A. Saxena, K. Thirunarayan, V. L. Shalin, Amit P. Sheth, Principles for Developing a Knowledge Graph of Interlinked Events from News Headlines on Twitter, CoRR 2018

[13] TagMe, https://sobigdata.d4science.org/web/tagme/tagme-help, [Accessed 4th Jun. 2021].

[14] E. Iglesias, S. Jozashoori, D. Chaves-Fraga, Di. Collarana, Ma. Vidal, SDM-RDFizer: An RML Interpreter for the Efficient Creation of RDF Knowledge Graphs. CIKM 2020

[15] B. Harandizadeh, S. Singh, Tweeki: Linking Named Entities on Twitter to a Knowledge Graph. W-NUT@EMNLP 2020:

[16] A. Dimou, M. V. Sande, P. Colpaert, R. Ver- borgh, E. Mannens, and R. V. de Walle. RML: A generic language for integrated RDF mappings of heterogeneous data. In Proceedings of the Workshop on Linked Data on the Web co-located with WWW, 2014.

[17] U. S ims ek, E. Kärle, and D. Fensel. RocketRML-A NodeJS implementation of a use-case specific RML mapper. In Pro- ceeding of the First International Workshop on Knowledge Graph Building, 2019.

[18] P. Ferragina, U. Scaiella, TAGME: on-the-fly annotation of short text fragments (by wikipedia entities). CIKM 2010: 1625-1628

[19] Santipantakis, Georgios M., Konstantinos I. Kotis, G. Vouros and C. Doulkeridis. “RDF-Gen: Generating RDF from Streaming and Archival Data.” Proceedings of the 8th International Conference on Web Intelligence, Mining and Semantics (2018): n. pag.

[20] Stanford OpenIE, `https://github.com/philipperemy/Stanford-OpenIE-Python`, [Accessed 7th Jun. 2021].