

Analyzing and Visualizing Twitter Streams based on Trending Hashtags

Bachelor Thesis

by

Manuel Kaschura

Degree Course: Industrial Engineering and Management

Institute for Applied Informatics and Formal
Description Methods (AIFB) in cooperation with Leibniz-
Institute for Information Infrastructure (FIZ - Karlsruhe)
KIT Department of Economics and Management

Advisor:	Prof. Dr. Harald Sack
Second Advisor:	Prof. Dr.-Ing. J. Marius Zöllner
Supervisor:	Dr. Mehwish Alam
Submitted:	30. December 2020

Abstract

Social networks are an incredibly large source of data. Millions of new data is generated every day when people from all over the world voluntarily share their thoughts and feelings about different topics. Information obtained from this data can be of high value for marketing, psychology, political science, etc. However, as millions of new data are generated every day, automated analysis, like Natural Language Processing, is required to extract information from this data on a large scale.

In this thesis, a new tool named Apollo, which analyzes Twitter streams for their sentiments and emotions and visualizes the results on an interactive globe visualization, is introduced. Therefore, Natural Language Processing foundations are explained, this includes sentiment analysis and emotion detection, as well as frame semantics. Furthermore, the Semantic Web is explained, as the analysis is based on knowledge graphs using Semantic Web technologies. The tool is applicable to English tweets with any content, however, because of the actual COVID-19 pandemic, only Tweets with the following keywords are considered: “COVID”, “corona”, and “coronavirus”. In the end of the thesis, sentiment analysis and emotion detection results are presented and interpreted.

Zusammenfassung

Täglich werden Millionen von Daten in sozialen Medien generiert. Menschen aus der ganzen Welt teilen freiwillig ihre Gedanken und Gefühle zu verschiedensten Themen. Informationen, die aus diesen Daten gewonnen werden, können beispielsweise für Marketing, Psychologie oder Politik nützlich sein. Da jeden Tag viele neue Daten generiert werden, ist eine automatisierte Analyse, z.B. mit Hilfe von *Natural Language Processing*, erforderlich, um Informationen aus diesen Daten zu gewinnen.

In dieser Arbeit wird ein neues Tool namens Apollo vorgestellt, das Twitter-Streams auf ihre Gefühle und Emotionen hin analysiert und die Ergebnisse auf einem interaktiven Globus visualisiert. Dazu werden zuerst die Grundlagen von *Natural Language Processing* erläutert. Dies beinhaltet *Sentiment Analysis* und *Emotion Detection*, sowie *Frame-Semantics*. Darüber hinaus wird das *Semantic Web* erläutert, da die Analyse auf Wissensgraphen unter Verwendung von *Semantic Web* Technologien basiert. Das Tool ist anwendbar auf englische Tweets mit beliebigem Inhalt, allerdings werden aufgrund der aktuellen COVID-19-Pandemie nur Tweets mit den folgenden Schlüsselwörtern berücksichtigt: "COVID", "corona" und "coronavirus". Am Ende der Arbeit werden die Ergebnisse der *Sentiment Analysis* und der *Emotion Detection* präsentiert und interpretiert.

Table of Content

Abstract	iii
Zusammenfassung	iv
List of Abbreviations	vii
List of Figures	viii
List of Tables	ix
1. Introduction	1
1.1. Motivation and Background	1
1.2. Objective	2
1.3. Structure of the Thesis	2
2. Foundations	3
2.1. Natural Language Processing	3
2.1.1. Lexical Resources	5
2.1.2. Word Segmentation	6
2.1.3. Word Sense Disambiguation.....	7
2.1.4. Frame Semantics.....	9
2.2. Semantic Web.....	10
2.2.1. Resource Description Framework.....	11
2.2.2. Knowledge Bases and Knowledge Graphs	12
2.2.3. Linguistic Linked Open Data.....	14
3. State-of-the-Art	16
3.1. Sentiment Analysis and Emotion Detection	16
3.2. Visualization.....	17
4. Methodology	18
4.1. Data Collection.....	18
4.2. Preprocessing	19
4.2.1. Text Preprocessing	19
4.2.2. Location Information.....	20
4.3. Sentiment Analysis	24
4.4. Emotion Detection.....	25
4.5. Frame Detection	26
4.6. Visualization.....	26
5. Results	29
6. Conclusion and Future Work	31
6.1. Summary	31

6.2. Future Work32

Declaration about the Thesis33

References34

List of Abbreviations

AI	Artificial Intelligence
ED	Emotion Detection
FD	Frame Detection
FE	Frame Element
KB	Knowledge Base
LKB	Lexical Knowledge Base
LLOD	Linguistic Linked Open Data
LOD	Linked Open Data
LU	Lexical Unit
NLP	Natural Language Processing
pos	part-of-speech
RDF	Resource Description Framework
RDFS	Resource Description Framework Schema
SA	Sentiment Analysis
SW	Semantic Web
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
WSD	Word Sense Disambiguation
WWW	World Wide Web

List of Figures

Figure 1 - Homonyms	4
Figure 2 - Synonyms	5
Figure 3 – Graph Representation of a Triple	11
Figure 4 – Linked Open Data Cloud	13
Figure 5 – DBpedia Ontology Type Structure	14
Figure 6 – Framester Cloud.....	15
Figure 7 – Distribution of Location Information	20
Figure 8 – Polygon Box Example.....	21
Figure 9 – Polygon Box Center	21
Figure 10 – Geopy.....	22
Figure 11 – Success of Nominatim Mappings.....	23
Figure 12 – Flowchart Stream Analysis	23
Figure 13 – Sentiment Analysis	27
Figure 14 – Emotion Detection	28
Figure 15 – Distribution of Tweets.....	29
Figure 16 – Distribution of Emotions.....	30

List of Tables

Table 1 – Lesk algorithm Example.....	8
Table 2 – Example Smileys.....	25
Table 3 – Color codes SA.....	28
Table 4 – Color codes ED	28

1. Introduction

Social networks are seemingly an infinite source of data. The information obtained from this data can be of high value for marketing, psychology, political science, etc. [1, 2] However, as millions of new data are generated every day, automated analysis is required to extract information from this data on a large scale. Natural Language Processing (NLP) and subfields e.g. Sentiment analysis (SA) and emotion detection (ED) are automated ways to extract meanings out of text documents or voice recordings. Applied to posts harvested from social media platforms they can be used to extract sentiments and emotions of people about certain events such as immigration, political elections, pandemics, etc. Furthermore, visualizing the results has proven to ensure traceability [3]. This thesis focuses on introducing the tool Apollo¹ which analyzes Twitter streams based on trending hashtags regarding the coronavirus. Apollo was created during this thesis. Geotagged Tweets, published in English, are analyzed for their sentiments and emotions using lexical resources and knowledge graphs. After performing the SA and ED, a globe visualization will be used for making the results more intuitive.

1.1. Motivation and Background

During the preparations of this study, in summer 2020, the coronavirus became a pandemic with global consequences, both economically and sociologically. Citizens must observe distance rules, shops are closed and in many countries around the world, it is mandatory to wear a mask while visiting public places such as supermarkets, restaurants, shopping centers, etc. Consequently, it is likely that people suffer in terms of mental health and connect negative emotions with the virus. The number of infections and deaths widely differ across countries², and every country handles the situation differently.

To compare the general feeling about COVID-19 in the countries, it promises interesting results to analyze the sentiments and emotions of people and then visualize the results of this analysis. As the sentiments and emotions differ depending on the location, geotagged data is needed. A big source of geotagged data are social media channels where people voluntarily share their location while they openly express their thoughts and feelings about trending topics.

Many social media platforms provide APIs which can be used to analyze aggregated user data. Not all APIs are suitable for analyzing purposes as they have restrictions in the number of requests a server can make to the given API. Due to the microblogging format of posts with up to 280 characters, Tweets are

¹ <http://covid-twitter-stream.fiz-karlsruhe.de/>

² <https://www.statista.com/statistics/1043366/novel-coronavirus-2019ncov-cases-worldwide-by-country/>

harder to process due to short or missing context and slang language. However, the API of Twitter has little restrictions in server requests and is thus suitable [6].

Before conducting the data analysis, it is good to have an overview of possible approaches to analyze posts about sentiments and emotions on a large scale and the different tools to visualize geotagged data.

1.2. Objective

The objective of this thesis is to explore and evaluate different algorithms and tools for analyzing geotagged Tweets about their sentiments and emotions. Furthermore, tools for visualization on a world map are explored and evaluated. Those tools will rely on the geotag information extracted from Tweets. As this study focusses on non-machine learning techniques the evaluated algorithms and tools rely on lexical resources and Semantic Web (SW) technologies. A further purpose of the work is to explain the methodology used to create the tool, Apollo.

1.3. Structure of the Thesis

After the introduction, the necessary background information is provided. In the third chapter, the current state-of-the-art in terms of sentiment analysis and emotion detection regarding coronavirus related Twitter-posts (Tweets) are presented. Moreover, different possibilities of visualizing data are discussed. In chapter four, *Methodology*, we explain the tool Apollo which has been developed during the thesis and highlight its used components. Afterward, the results of the tool are highlighted. The last chapter concludes the thesis. This includes implications for future research as well as limitations of this study.

2. Foundations

This chapter provides fundamental background information relevant for this thesis. First, NLP with a focus on non-machine learning options is explained. This contains Lexical Resources for NLP and explains Word Sense Disambiguation (WSD) as well as Word Segmentation. Afterward, the Semantic Web, Linked Open Data (LOD) and Knowledge Graphs, Linguistic Linked Open Data (LLOD), and DBpedia are explained.

2.1. Natural Language Processing

Natural Language Processing is a subfield of computational linguistics and Artificial Intelligence (AI). It describes the process of automated information extraction out of text documents or voice inputs. It can be used for machine translation, summarization, speech recognition, topic segmentation, sentiment analysis, emotion detection, etc. NLP is an approach of computers to analyze, understand, and derive meaning from natural languages. [7]

Some of these tasks might be hard for humans as well, e.g. learning new languages or understanding the right emotions out of text documents. However, for humans it is quite easy, if a context is given, to distinguish between two words which are Homonyms. Homonyms are words that are pronounced and/or written the same way but have different meanings. For example, the word “bank”. Consider two sentences: “You can withdraw money from the bank” and “The river overflowed the bank”. For a human reader, it is clear that the first sentence refers to a bank as a financial institution whereas in the second sentence a sloping bank that is near water is meant (see figure 1). For computers, this task appears to be difficult, because to get the right sense of a word, besides lexical and grammatical information, knowledge about semantic, pragmatic, and general world knowledge must be available [7]. For NLP there are supervised and unsupervised machine learning and statistical methods as well as methods based on lexical resources. This study focuses on the latter, lexical resources.

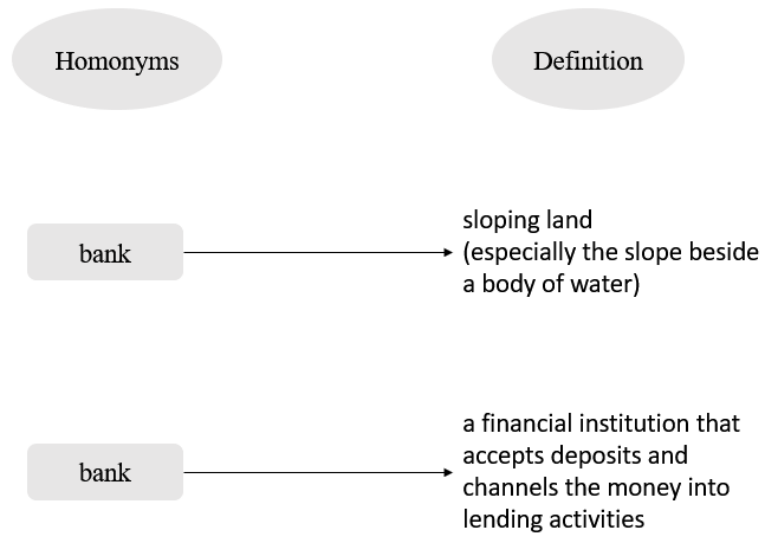


Figure 1 - Homonyms

2.1.1. Lexical Resources

The idea behind lexical resources is to use dictionaries that provide different background information about linguistics and semantics for words. The information is stored in databases. When analyzing a text, it is often the case that different words have the same meaning. Synonyms make it harder for computers to fulfill NLP because the computer must know that each different word representation of a synonym has the same meaning. Instead, in a preprocessing step, all synonyms could be summarized to one formal representation of the synonym. This representation is a set of synonyms and is called synset. In a next step, meaning or definitions could be added to each synset. The following synonyms clarify it further: aeroplane, plane, airbus, airplane can all be defined as “an aircraft that has a fixed wing and is powered by propellers or jets”³ and could be mapped to the synset “airplane” (see figure 2). A preprocessing step when analyzing texts maps each word to its synset representation.

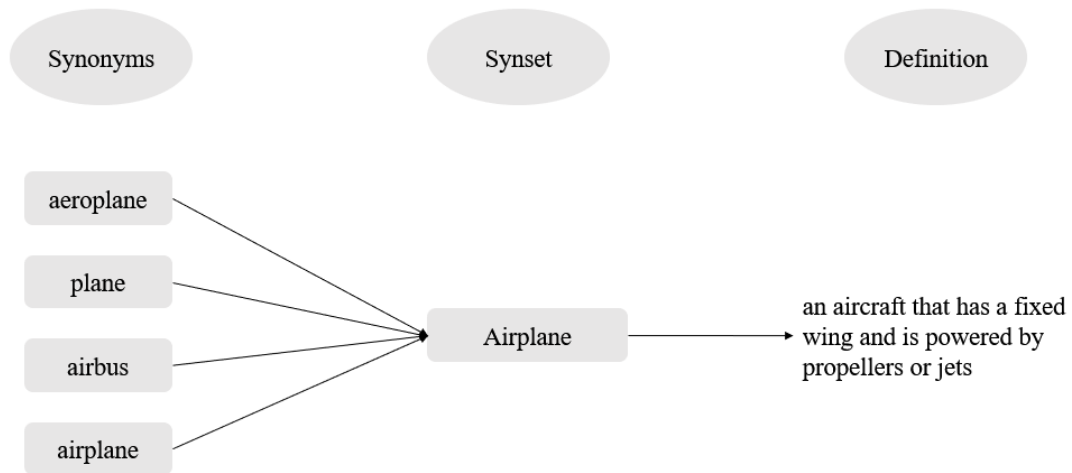


Figure 2 - Synonyms

WordNet

A well-known lexical resource used in NLP is *WordNet*. WordNet is a large dictionary of the English language [9]. Rather than alphabetically arranged, WordNet is arranged semantically. It groups synonymous words such as nouns, verbs, adjectives, and adverbs into synsets, where each synset expresses a different word sense. In WordNet, 117.000 synsets are interlinked through conceptual-semantic and lexical relations.

³ <http://wordnet-rdf.princeton.edu/lemma/airplane>

For example, if one searches for 'dog', WordNet provides a list of synsets which are related to 'dog':

```
Synset('dog.n.01')  
Synset('frump.n.01')  
Synset('dog.n.03')  
Synset('cad.n.01')  
Synset('frank.n.02')  
Synset('paw1.n.01')  
Synset('andiron.n.01')  
Synset('chase.v.01').
```

Each synset has a different definition. E.g. *dog.n.01* is defined as '*a member of the genus Canis (probably descended from the common wolf) that has been domesticated by man since prehistoric times; occurs in many breeds*' whereas *dog.n.03* is defined as '*informal term for a man*'. In contrast, the definition of the synset *chase.v.01* is '*go after with the intent to catch*' which has a relation to dogs but has no explicit connection to them (the word dog does not appear in its definition). Note that the synsets are structured in a descendant order, the first synset is the one which appears most frequently in lexical corpora.

2.1.2. Word Segmentation

Word Segmentation becomes necessary when there are missing space delimiters between words. It is an initial step for many NLP tasks, such as part-of-speech (pos) tagging, machine translation, or sentiment analysis [10]. Word Segmentation means segmenting the words by adding the missing space delimiters. In English texts, space delimiters initially exist, whereas in other languages such as Cambodian (Khmer) no space delimiters are used [11]. However, in English, when applying voice recognition or when processing social media data Word Segmentation must be applied as well. Social media posts often come with hashtags (e.g. #themenarehere) which do not have space delimiters between words. To be able to further process the data, space delimiters must be added (# the men are here). A possibility to segment text is to scan each character one at a time and lookup those characters in a dictionary. As soon as the chain of characters matches a word in the dictionary the sequence is segmented as a word.

However, this sometimes causes the problem of matching shorter length words. Consider segmenting the word "theme" it would result in "the" and "me" even if "theme" is meant. There are certain ways to

improve this approach. A way to avoid matching the shortest word is a maximal matching algorithm which is a greedy algorithm that matches the longest word. In the example above (#themenarehere) the algorithm finds the words “the”, “them” and “theme” and segments at the longest word (theme). This would result in “theme narethere” which does not make sense either. To solve this issue a suggestion is to match backward as well.

The approach by Narin Bi et. al [12] is known as bi-directional maximal matching. The algorithm not only works its way from left to right (forward matching) but also from right to left (backward matching) to then choose the best result. Narin Bi et al. shows an accuracy of 98% for the bi-directional maximal matching algorithm [12].

Another approach to solve the issues with the greedy maximal matching algorithm is called “maximum matching”. This algorithm by Phylypo Tum [13] segments the sentence by multiple possibilities and chooses the one with the least amount of words.

2.1.3. Word Sense Disambiguation

If words are spelled the same and sound the same but have different meanings it is necessary to get the right meaning according to the context. WSD is the process of selecting the right meaning for a word, based on the given context [14]. Different features like pos, morphology, verb-object relationships, and lexical features should be considered when applying WSD [15, 16]. However, combining all those features is a difficult task so different approaches address different features. There are several approaches to disambiguate words such as supervised methods, semi-supervised methods, unsupervised methods, or methods based on lexical resources. Supervised methods use a (manually annotated) training set to train the machine learning algorithm. Semi-supervised methods use small annotated corpora or word-aligned bilingual corpora. Unsupervised methods work without external information and use unannotated corpora.

For lexical resources, a prerequisite is a lexicon of potential word senses. The Lesk algorithm is a famous algorithm applied to lexical resources [17]. The first implementation of the Lesk algorithm works based on classical lexica such as Oxford Advanced Learner’s. It compares the definitions, or *glosses*, of homonyms with glosses of other homonyms in the same sentence and counts the overlaps within those glosses. The word sense that has the largest number of overlaps from its glosses with the glosses of other words is assigned to that word. Consider the following example.

How to tell a computer the difference between a *pine cone* and *ice cream cone*? The possible glosses are illustrated in Table 1.

Table 1 – Lesk algorithm Example [17]

<i>word</i>	<i>gloss</i>
pine	1 kinds of evergreen tree with needle-shaped leaves
	2 waste away through sorrow or illness
cone	1 solid body which narrows to a point
	2 something of this shape whether solid or hollow
	3 fruit of certain evergreen trees

The biggest overlap of words in the glosses are between *pine-1* and *cone-3*. Namely *evergreen* and *tree*. The Lesk algorithm returns the senses *pine-1* and *cone-3*.

Satanjeev Banerjee and Ted Pedersen [14] adapted the Lesk algorithm to use the advanced NLP dictionary WordNet [9] instead of classical dictionaries. It scans all glosses for each possible synset and compares it with the glosses of neighboring synsets. Measuring the accuracy of the adapted Lesk algorithm using the English SENSEVAL-2⁴ lexical sample data, they achieved an overall accuracy of 32% versus 16% with the traditional Lesk approach [14].

Some other approaches for WSD are Babelfy⁵ and UKB⁶. *Babelfy* is a unified and graph-based approach to entity linking and WSD available in 271 languages. It identifies possible meanings and is coupled with a graph heuristic which selects high-coherence semantic interpretations. It is based on BabelNet⁷ which is a multilingual semantic network and performs WSD and entity linking. [22]

UKB is a tool used for graph based WSD and lexical similarity. More precisely, it is a collection of tools, including tools to produce graphs from knowledge bases like WordNet. UKB applies the personalized PageRank [23] and uses the knowledge in Lexical Knowledge Bases (LKBs) to rank the vertices of the LKB and thus performs disambiguation.

⁴ <http://www.hipposmond.com/senseval2/>

⁵ <http://babelfy.org/>

⁶ <https://ixa2.si.ehu.es/ukb/>

⁷ <https://babelnet.org/>

2.1.4. Frame Semantics

The first research in the field of semantics and linguistics by Fillmore was *case grammar* [24]. He proposed a set of hierarchical structured semantic roles (often referred to as *deep cases*) such as Dative, Instrumental, Agentive, Locative, and Objective that are used for identifying grammatical functions. At the time Fillmore's approach differed from other approaches in that they explicitly required the identification of a limited set of semantic roles that could be applied to any argument of any verb. However, further research showed limitations and problems of case grammar as discussed in [25].

Firstly developed by Fillmore, Frame Semantics is a theory that builds upon case grammar. He claims that to be able to understand words the right way, one needs essential background knowledge about the word [26]. It connects linguistic semantics with general knowledge.

A common example is the Commercial Transaction Frame that consists of a buyer, a seller, goods, and money. The strongest semantic links that connect to this frame are *buy, sell, pay, spend, cost, and charge*. Each individual frame element (FE) indicates or evokes different aspects of the frame. The verb *buy* is about the buyer and the goods, with the background of the seller and the money; *Sell* is about the seller and the goods, backgrounding the buyer and the money; and so on. Without knowing commercial transactions and without knowing the meaning of any of those verbs it is not possible to fully understand the situation and to understand the meaning of any one of these verbs the right way. The verbs must be described as holistic. This includes grammatical information and different patterns in which they occur. For example, in the sentence *Peter bought the smartphone from Sarah for 100\$*, the subject, *Peter*, is the buyer, and the direct object, *the smartphone*, is the goods; both FEs are needed for a proper sentence. The optional backgrounded elements are the seller, *from Sarah*, and the money, *for 100\$*. [27]

Fillmore and Atkins summarized Frame Semantics in the following quote:

“A word's meaning can be understood only with reference to a structured background of experience, beliefs, or practices, constituting a kind of conceptual prerequisite for understanding the meaning. Speakers can be said to know the meaning of the word only by first understanding the background frames that motivate the concept that the word encodes.” [28]

FrameNet

After the theory of Frame Semantics had been established, applying those theoretical principles into a form that could be machine-usable was still missing. *FrameNet*⁸, a project of the International Computer Science Institute (ICSI) in Berkeley, was developed for this purpose. It is a lexical resource, which provides semantic and syntactic properties of words, in particular their meaning, the combination of their syntactic and semantic alternatives, and their relation to the semantic frames which determine their meanings. Pairings of words and their meanings, which is usually a single word in a given frame is called Lexical Unit (LU). In the FrameNet corpora, each LU links to a particular Semantic Frame that it evokes.

For example, cooking usually involves someone cooking (*Cook*), the nutrition that is to be prepared (*Food*), and a source of heat (*Heating_instrument*). This is represented as the frame called *Apply_heat*, and the *Cook*, *Food*, and *Heating_instrument* are the FEs. Words that evoke this frame, such as *fry*, *boil*, and *bake* are called LUs of the *Apply_heat* frame. However, there are some more complex frames, such as *Revenge*, which involves more FEs (*Offender*, *Injury*, *Injured_Party*, *Avenger*, and *Punishment*). FrameNet defines the frames and annotates sentences, as in the following examples of *Apply_heat* and *Revenge*:

- ... [*Cook* the boys] ... GRILL [*Food* their catches] [*Heating_instrument* on an open fire].
- [*Avenger* I] 'll GET EVEN [*Offender* with you] [*Injury* for this]!

2.2. Semantic Web

The Semantic Web is an extension of the World Wide Web (WWW). It was invented by Tim Berners-Lee, the inventor of the WWW. In the year 2001, he claimed that the current WWW has developed as a human-only readable database, as uploaded content does not share any certain structure.

To make the WWW machine-readable the idea is to store all information on the WWW in a standardized format. According to Tim Berners-Lee, this will result in an infrastructure that enhances the development of automated Web services [30]. The SW facilitates machines the ability to understand the meaning (semantics) of the information on the WWW [31].

⁸ <https://framenet.icsi.berkeley.edu/fndrupal/WhatIsFrameNet>

2.2.1. Resource Description Framework

The *Resource Description Framework*⁹ (RDF) is the key component of the SW. It enables the representation of knowledge in a structured and machine-understandable way. RDF is the standard format for encoding data in the SW and is used for representing information about (real-world) entities and their relations. RDF breaks down information into facts so that each fact has a clearly defined form.

The facts in RDF are represented in triple statements. Each triple consists of:

<subject, predicate, object>

Subjects and properties are represented via a unique address, the Universal Resource Identifier (URI). Objects can be either URIs or literals. The latter is the case if it describes data values that do not have a URI (e.g. a year or date or any value which has no separate entity). Examples for URIs are Uniform Resource Locators (URLs) for Web pages or ISBN and ISSN for books. A Resource can be any clear identity e.g. web pages, persons, relations among objects, etc. E.g. “Albert Einstein educated at ETH Zurich”. “Albert Einstein” is the subject, “educated at” the predicate and “ETH Zurich” the object.

A further enhancement of RDF is the *RDF Schema*¹⁰ (RDFS). RDFS is a way of adding semantics to the RDF data. It is a knowledge representation language that can be used to describe classes, sub-classes, and properties of RDF resources [31]. It allows to not only describe resources and relations between them but grants the possibility of adding meaning.

The according RDF triple for the example would be:

`<http://dbpedia.org/resource/Albert_Einstein> <http://dbpedia.org/property/education> <http://dbpedia.org/resource/ETH_Zurich>`

A possible way of representing triples is by using a directed graph (see figure 3).

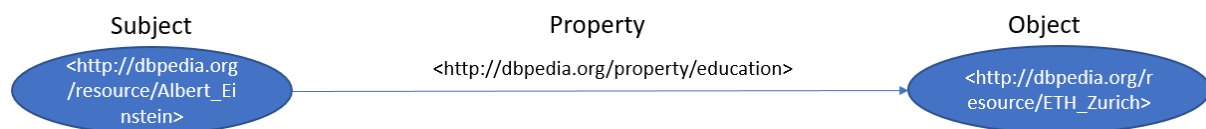


Figure 3 – Graph Representation of a Triple

⁹ <https://www.w3.org/TR/rdf-primer/>

¹⁰ <https://www.w3.org/TR/rdf-schema/>

2.2.2. Knowledge Bases and Knowledge Graphs

“A *knowledge base* (KB) is a structured knowledge repository that contains a set of facts (assertions) about entities” [34]. A knowledge repository is a collection of entities with different information about entities such as descriptions or properties. A classic example of a knowledge repository is Wikipedia. Knowledge Bases provide a structured collection of data. They add semantic meaning to the data, which contains a formal representation with classes, relations, and instances (ontologies and dictionaries) as well as defaults to interpret the data. Examples of knowledge bases are DBpedia, YAGO, Wikidata, etc. We further explain a knowledge base with the example of DBpedia in the chapter below.

If we expand the example of figure 3 by connecting information and link data over the WWW this results in a *knowledge graph*. A knowledge graph is a knowledge base ordered as a graph used to store linked information about entities. Nodes represent subjects and objects and edges represent predicates.

The combination of knowledge graphs with open data structured in RDF triples and representable as a graph is called Linked Open Data (LOD, see figure 4). It allows structured queries via the RDF Query Language SPARQL. That way Wikipedia information becomes machine-accessible.

The nucleus of LOD is DBpedia which extracts and provides structured information from Wikipedia. This revolutionizes the accessibility of information in the WWW. Assume one wants to develop an intelligent personal assistant agent (e.g. Amazon Alexa). To answer all questions asked by the end customer it is necessary to provide information found in the WWW. But instead of searching for information in documents the agent could use a SPARQL query and provide the received information to the end-user. A possible query could be: list all cities in Germany with more than 20,000 inhabitants.

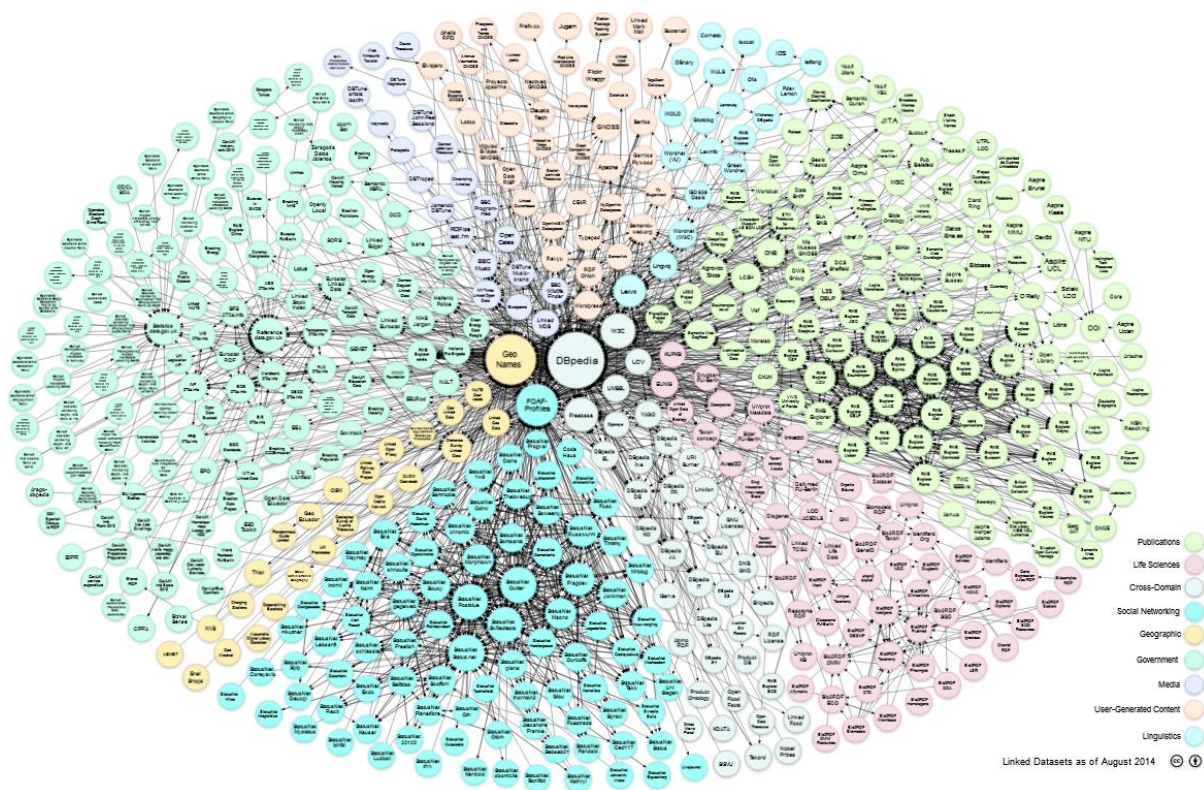


Figure 4 – Linked Open Data Cloud ¹¹

DBpedia

*DBpedia*¹² is a community project that extracts multilingual and structured knowledge from Wikipedia and provides it freely available on the WWW using SW and Linked Data technologies. It provides knowledge from 111 different language editions of Wikipedia. The English DBpedia knowledge base is the largest with over 400 million facts that describe 3.7 million things [37]. The extraction is done automatically through an open-source extraction framework [38]. While Wikipedia is more of a human-readable knowledge repository, DBpedia can be seen as the “machine-readable Wikipedia”. It allows structured queries via a SPARQL endpoint.

¹¹ <https://lod-cloud.net/>

¹² <https://wiki.dbpedia.org/>

The structural fundament of DBpedia is defined by the *DBpedia Ontology*. It creates properties and classes which structure the resources (see figure 5). The DBpedia Ontology was created manually using the most frequently used infoboxes in Wikipedia.

In short, DBpedia is a large collection of RDF graphs with predefined semantics that provide machine-readable data extracted from Wikipedia.

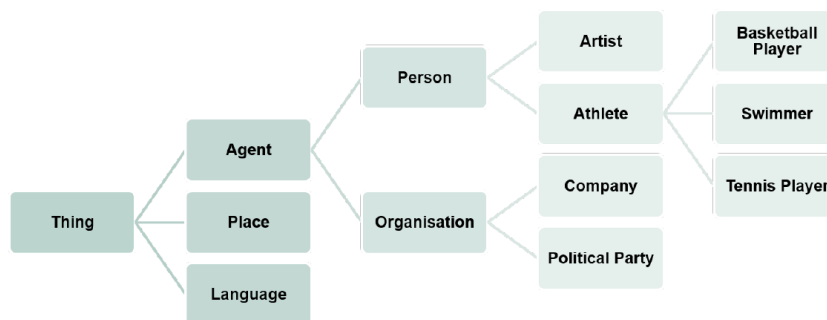


Figure 5 – DBpedia Ontology Type Structure [39]

2.2.3. Linguistic Linked Open Data

*Linguistic Linked Open Data*¹³ (LLOD) is the idea of combining linguistics and NLP with Linked Open Data. LLOD should be openly available and the elements should be uniquely identified via URIs. That means that LLOD resources use web standards of the LOD such as RDF. Links to other resources are very important as they build the foundations of semantic relations and help discover new resources.

There are many different approaches for LLOD but not always they are interoperable. For an overview of different LLOD applications and their links see the LLOD Cloud¹⁴. The LLOD Cloud represents a temporal snapshot of linguistic datasets published on the WWW. It is maintained by the Open Linguistics Working Group¹⁵ (OWLG), which is an interdisciplinary network where any individual dedicated to computational linguistic and linguistic resources can participate. [43]

¹³ <https://linguistic-lod.org/>

¹⁴ <https://lod-cloud.net/clouds/linguistic-lod.svg>

¹⁵ <https://linguistics.okfn.org/>

Framester

Framester [44] is a hub between different linguistic resources. It provides a new set of linked linguistic resources and is based on Frame Semantics. Framester provides new mappings between FrameNet, WordNet, and more.

The interoperability between the used resources increases because Framester standardizes the predicate spaces. It is accessible via a SPARQL endpoint. Some of the other linguistic resources are SentiWordNet and DepecheMood. For example, there has not been a link between DepecheMood and WordNet, but Framester closed this gap. For a full overview of all linked resources see figure 6.

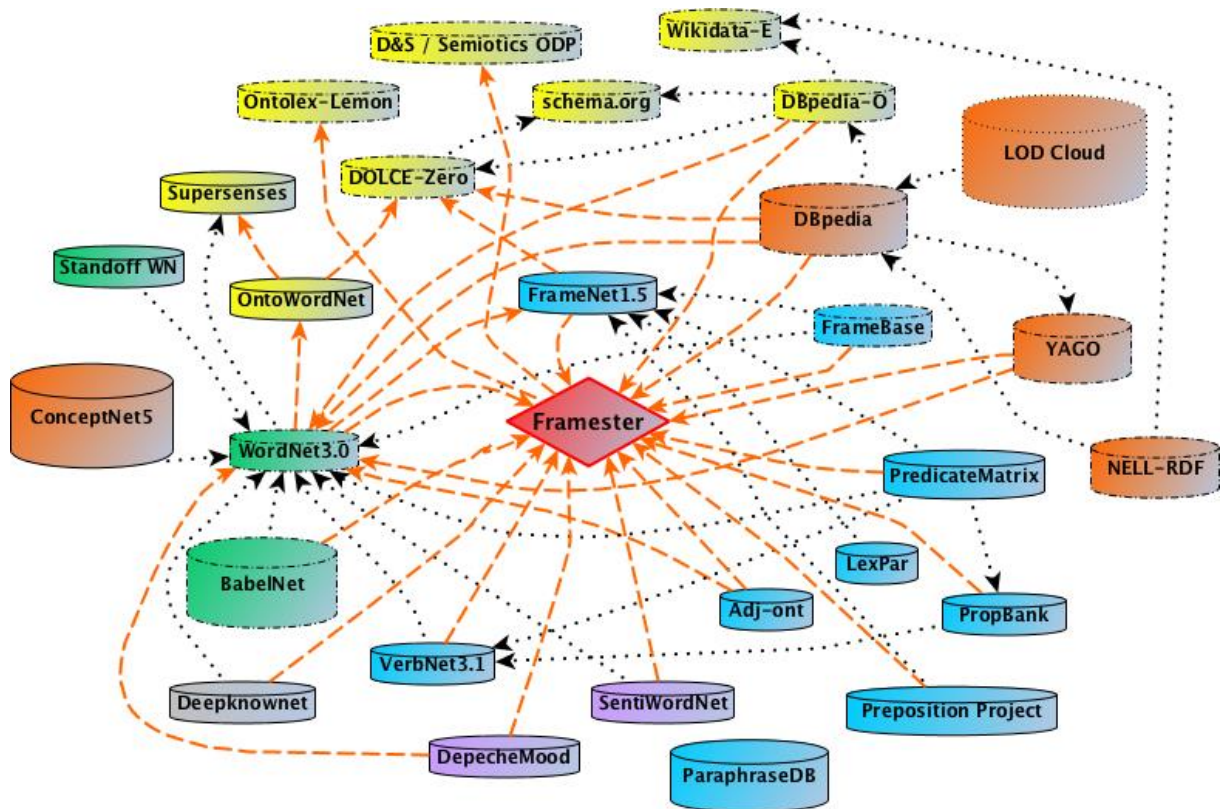


Figure 6 – Framester Cloud [44]

3. State-of-the-Art

Current State-of-the-Art heavily relies on lexical resources. Short information about used lexical resources is provided at the beginning of each subchapter. First, we will talk about State-of-the-Art in SA and then in ED. Afterward, the current possibilities of data visualization are highlighted. As ED is a further development of SA many State-of-the-Art technologies rely on the same lexical resource, *WordNet*.

3.1. Sentiment Analysis and Emotion Detection

SA and ED belong to the field of natural language processing (NLP). This topic has already been researched since the year 2000 [45] [46]. Although there are different approaches to tackle NLP tasks, we provide brief information about four papers on this topic. As a prerequisite, one should be aware of SentiWordNet. SentiWordNet is a dictionary for sentiment analysis [47]. It provides mappings between the synsets of WordNet and positive and negative scores. Each score is a floating number between zero and one.

A paper published in 2015 introduces a frame-based SA, Sentilo [48]. Sentilo performs SA by combining NLP techniques with SW technologies. Sentences, which are biased towards an opinion (positive or negative) are provided with formal representations in form of resource description framework (RDF) graphs. In the RDF graph, the characterizing concepts and relations of opinion sentences are defined. What distinguishes Sentilo from other SA approaches is that in opinioned sentences it is capable of detecting the main topic and if available the subtopic. That way, sentences with a negative opinion but with explicit positive verbs in it can be rightly classified as negative, e.g. “People hope that the President will be condemned by the judges”. The verb “hope” has a positive tone, although the overall opinion towards the object (“the President”) is negative.

Emily Chen et. al introduced the first public coronavirus twitter dataset [49]. They collected multilingual tweets about the coronavirus and gathered about 450GB of raw data (around 50 million tweets) from January 22, 2020, until March 16, 2020, using Twitter’s streaming API and Tweepy¹⁶.

Dimitar Dimitrov et. al created a knowledge graph of Tweets about the coronavirus and its impact on society [51]. Furthermore, they updated the TweetsKB¹⁷ dataset and pipeline. The pipeline now uses the April 2020 Wikipedia dump to perform entity linking.

¹⁶ <https://github.com/tweepy/tweepy/>

¹⁷ <https://data.gesis.org/tweetskb/>

Irene Li et. al applied NLP techniques to analyze tweets regarding the coronavirus towards the mental health of people [53]. Therefore, they trained deep models to classify tweets into eight different emotions to then find relations between the emotions sadness and fear and their causing keywords. To build a dataset to train the deep models with, they manually labeled 1000 English tweets.

3.2. Visualization

There are limited possibilities to visualize geotagged data. Since the location information should be preserved, we plot the data on a map. Either the data can be plotted on a world map or more specific areas on earth e.g. Europe.

Tweets of the Nation [54] is an interactive Tweet visualization tool that enables users to observe the most popular hashtags posted in the last 24 hours in any country of the world.

Eduardo Duarte et. al implemented *Living Globe*, a three-dimensional interactive visualization of world demographic data [55]. For the user, it is possible to explore demographic data on a globe visualization. However, the idea is to further develop the API, so the input data is no longer restricted to demographic data.

Furthermore, there is the *WebGL Globe*¹⁸ which is a three-dimensional geographic data visualization tool created by the Google Data Arts Team. It is possible to visualize any data if one has longitude, latitude, and magnitude of a data point. The magnitude of the data point is visualized as bars on the interactive globe. A high magnitude results in a high bar and a low magnitude results in a small bar.

¹⁸ <https://github.com/dataarts/webgl-globe>

4. Methodology

The aim of this thesis is not only to summarize the most recent research topics but also to present the newly developed tool Apollo, which provides a practical implementation of the above-discussed possibilities. The goal is to analyze Twitter streams for their sentiments and emotions and visualize the results on a globe visualization. Only Tweets containing certain keywords should be collected. Due to the actual coronavirus pandemic, we picked the keywords “COVID”, “corona”, and “coronavirus”.

This chapter describes the used algorithms and tools and the general process sequence in the program. Chapter 4.1. Data Collection explains which kind of data got collected from where. Then, the preprocessing steps that are necessary for SA, ED, and the visualization are highlighted. This contains the preprocessing of both, the Tweet text, and the location information. The former needs to be done before performing the SA and ED, while the latter precedes the data visualization. The resulting visualization is available online at FIZ¹⁹.

4.1. Data Collection

The tool Apollo collects social media data from Twitter. We analyze Tweets that contain the following keywords: “COVID”, “corona”, “coronavirus”. Those keywords were chosen because we expect that most of the Tweets regarding the coronavirus contain at least one of these three keywords. The Twitter API has few restrictions for free users, i.e. it is possible to stream around a maximum of 60 Tweets per second. All English Tweets containing the keywords are harvested. Only English Tweets are considered because NLP techniques such as SA and ED are most advanced for the English language, so we will leave multilingual analysis to future work.

To visualize the results of the analysis on a world map it is also mandatory that only Tweets with location information are kept for further processing. The free Twitter API does not allow a query filter for Tweets containing location information. Due to that, the Tweets are filtered for location information after streaming. Twitter provides three kinds of location information which will be further explained in Chapter 4.2.2. The Tweets come in the JSON format.

¹⁹ <http://covid-twitter-stream.fiz-karlsruhe.de/>

At the time of this writing, roughly 15 Tweets per second in the English language regarding coronavirus with location information are streamed. This results in approximately 900 Tweets per minute. Tweets are from many countries in the world; however, the global distribution strongly differs as we search for English Tweets only. Furthermore, in some countries, Twitter is not used as often or e.g. in China, the usage is prohibited by the government.

4.2. Preprocessing

Tweets contain lots of information²⁰, such as Tweet text, different URLs, location information, etc. The relevant information for the analysis are location information, timestamp (*created_at*), and the text of the Tweet (*text*). For the visualization, the location information and the results of the text analysis are needed. The timestamp is used, so that Apollo users know from which sliding time windows the Tweets are (see chapter 4.6. for more information).

4.2.1. Text Preprocessing

To fulfill the SA and ED a preprocessing of the text is necessary. The text contains at-mentions, hashtags, retweet abbreviations (RT), and Unicode representations of symbols, smileys, or emoticons. First, we scan the text for smileys and emojis and save them for SA later. Then, we remove the at-mentions, URLs, and numbers. Hashtags are handled as follows:

1. Remove the hashtag symbol ('#')
2. If necessary, segment the words using PyWSD²¹.

Next, we use PyWSD to get WordNet synsets for all words in the sentence. PyWSD is a python implementation that returns WordNet synsets after applying the Lesk algorithm for WSD. As there are no synsets for stopwords this part removes stopwords as well. WordNet synsets are necessary to analyze for sentiments and emotions.

²⁰ <https://developer.twitter.com/en/docs/tweets/data-dictionary/overview/intro-to-tweet-json>

²¹ <https://github.com/alvations/pywsd>

4.2.2. Location Information

Location information provided by Twitter can be either point coordinates (“Coordinates”), a polygon box (“Place”), or a string representing a location (“Location”), e.g. Houston, TX. According to Twitter²², one must first enable precise location on one’s device and then tap the location icon in the Tweet compose box. Then a list of places can be chosen from and the chosen one gets added to the Tweet. Depending on the chosen place, the Tweet comes with either “Place” or “Location” tag. The “Place” tag appears if a specific business, a landmark, or a point of interest is chosen whereas “Location” tag appears when a city is chosen. However, if one attaches a photo to a Tweet using the in-app camera and allows location information, the precise “Coordinates” (latitude, longitude) are added to the Tweet.

As for a proper visualization point coordinates are needed, the polygon box from the “Place” tag and the string representation from the “Location” tag must be transformed to point coordinates.

To get an idea of how the different location types are distributed across Tweets, we used a sample of 74310 Tweets collected over multiple Twitter streams. Most Tweets come with “Location”. “Place” and “Coordinates” combined are below one percent (see figure 7).

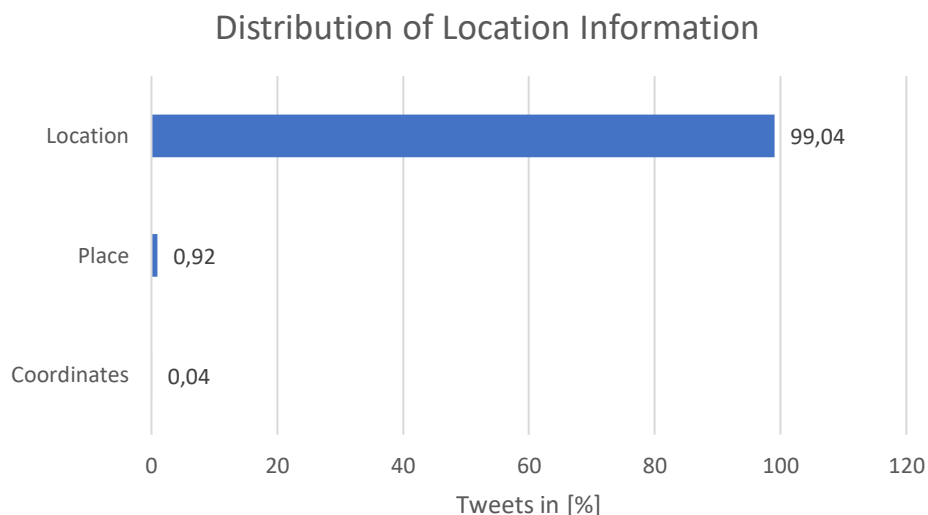


Figure 7 – Distribution of Location Information

²² <https://help.twitter.com/en/using-twitter/tweet-location>

Polygon Box

In the case of a returned polygon box, we use a mathematical function to get the center of the rectangle. The GPS information from a polygon box consists of four corners, clockwise starting at the bottom left corner (see figure 8 for an example).

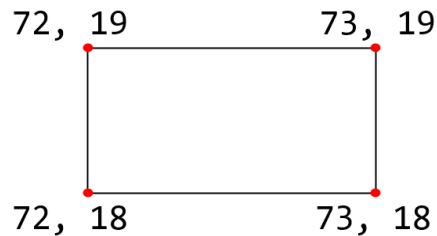


Figure 8 – Polygon Box Example
Possible return from Twitter: `[[[72, 18], [72, 19], [73, 19], [73, 18]]]`

Two corners are sufficient to get the center of the box. Assume two points (x, y) where $x = (x_1, x_2)$ represents the bottom left corner and $y = (y_1, y_2)$ represents the top right corner. The first value represents the longitude and the second value represents the latitude. To get the center of the box we need the center of longitude and the center of latitude coordinates:

$$longitude_{center} = \frac{x_1 + y_1}{2}$$

$$latitude_{center} = \frac{x_2 + y_2}{2}$$

See figure 9 for an example.

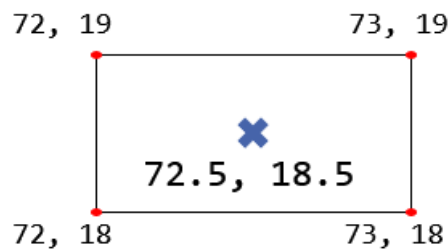


Figure 9 – Polygon Box Center
Returned Center: `[72.5, 18.5]`

String Representation

In the case of returned “Location”, we need to use an external service to transform the information to point coordinates. For this process, the Geopy library²³ has been used. Geopy is a library that wraps the APIs of 28 different geocoding services in Python (see Figure 10). Most free geocoding services have some kind of usage restriction. The most common restrictions are the number of requests per second, day, week, or month (around 3000 per month). As we get roughly 900 Tweets per minute and over 99% have “Location” as geotag 895 requests need to be made per minute which would result in 1,288,800 requests per day. For that reason, geocoding services with a maximum number of requests cannot be used.

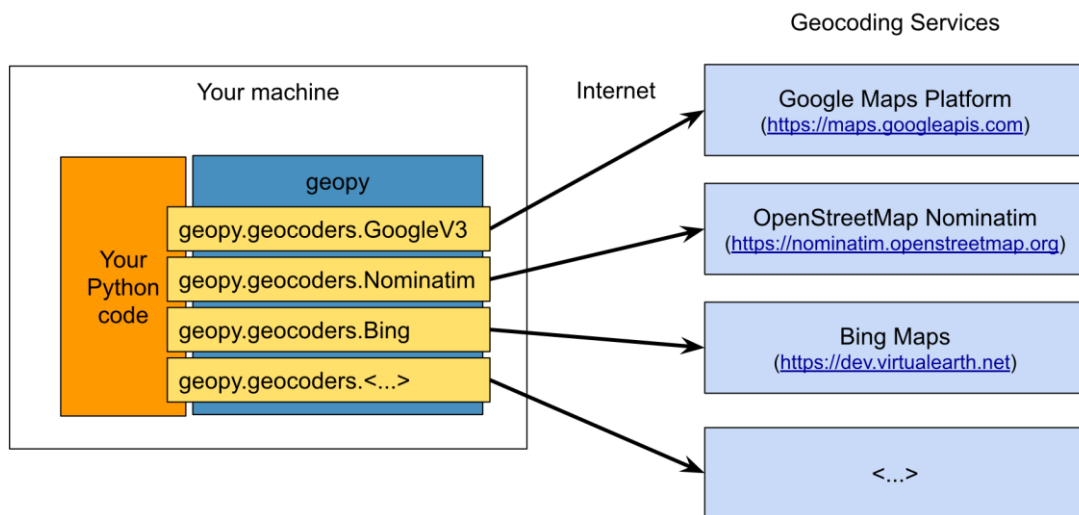


Figure 10 – Geopy²⁴

The only free geocoding service we found which is suitable for the task is *Nominatim*²⁵. Nominatim allows one request per second but has no other limits. Figure 11 shows that the majority of “Locations” are successfully mapped to point coordinates.

The main reason for unsuccessful mappings is that Twitter users themselves can create locations, thus sometimes there are non-valid “Locations” such as *On Six Continents* or *overthere*. Furthermore, sometimes the “Location” comes with abbreviations the geocoding service is not aware of.

²³ <https://pypi.org/project/geopy/>

²⁴ https://geopy.readthedocs.io/en/stable/_static/geopy_and_geocoding_services.svg

²⁵ <https://nominatim.org/>

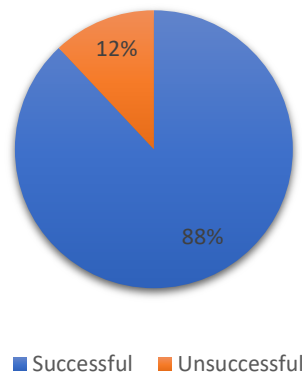


Figure 11 – Success of Nominatim Mappings

Still, the Nominatim restrictions of one request per second results in a time delay between the Twitter Stream and analysis and thus is the reason we cannot analyze in real-time. Instead, we stream for set time slides of ten minutes, starting the analysis in parallel. The stream stops after ten minutes and does not restart before the analysis is finished as well. The duration of the analysis is not consistent. It depends on the performance of the geocoding service, which varies within the day, and the internet connection of the server. However, the analysis takes approximately 12 times longer, so, ten minutes of streaming take around 120 minutes of analysis. The flowchart in figure 12 visualizes the sequence.

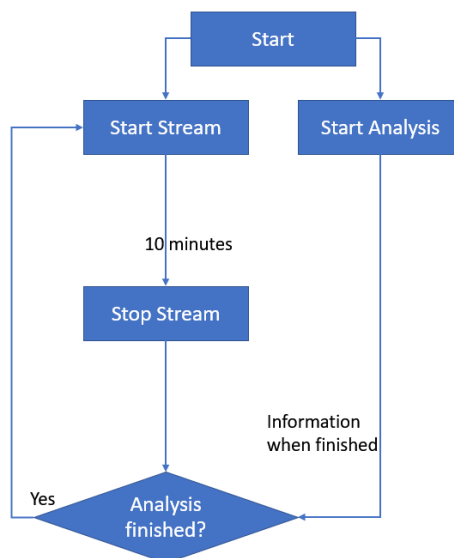


Figure 12 – Flowchart Stream Analysis

4.3. Sentiment Analysis

For SA the list of synsets returned from the preprocessing step is used. For each synset, we get positive and negative sentiments using SentiWordNet. We access SentiWordNet using Framester. However, it turned out that SPARQL queries are too slow for our use-case, so we decided to use *grep* instead. Grep allows very fast regular expression searches on plain-text data sets. It is applied to mappings that were provided by Framester and are available online²⁶. It returns a positive and a negative score for each synset.

For the visualization, a single value is needed that represents the height of the bar. Thus, for each sentence the score gets calculated as follows:

1. Sum up all negative scores
2. Sum up all positive scores
3. Subtract the negative sum of the positive sum
4. Divide by the number of synsets to get the average score
5. Return the average score
6. Derive a color index from the average score

For the visualization, a color index that maps to the color is needed. As we have the sentiments positive, neutral, and negative, three indices are sufficient. Zero represents positive, one represents neutral, and two represents negative Tweets. See table 3 in chapter 4.6. Visualization.



However, this changes if smileys were found in the preprocessing step. We manually annotated 112 smileys or, to be more precise, their Unicode representation to one category (positive, negative, neutral) by going through the first 182 smileys on this website²⁷. We assigned 11 to neutral, 46 to positive, and 55 to negative. These mappings are available online²⁸. For simplification reasons, we assume that all smileys have an equal influence on the sentiment. So, there is no distinction in the sentiment score between the two smileys in the table below.

²⁶ <https://github.com/ISE-FIZKarlsruhe/TwitterStreamAnalysis>

²⁷ <http://www.unicode.org/emoji/charts/full-emoji-list.html>

²⁸ <https://github.com/ISE-FIZKarlsruhe/TwitterStreamAnalysis>

Table 2 – Example Smileys

Smiley	Unicode
	U+1F642
	U+1F600

We set the positive and negative scores of smileys to 0.5. For comparison, in SentiWordNet the synset *happy* has a positive score of 0.75.

If there are more positive than negative smileys the sum of positive scores is increased and if there are more negative than positive smileys the sum of negative scores is increased. In either way, the sum is divided by the number of synsets plus one. If an equal number of positive and negative smileys is found, no changes are applied.

4.4. Emotion Detection

For ED we use *DepecheMood*. DepecheMood is a lexical resource for emotion detection which is derived from crowd annotated news. Framester provides mappings between WordNet synsets and DepecheMood emotion scores available via SPARQL queries. However, for the same reason as discussed in Chapter 4.3. we use grep. DepecheMood categorizes eight different emotions (afraid, amused, angry, annoyed, don't care, happy, inspired, sad). It returns eight emotion scores for each synset.

For the visualization, a single value that represents the height of the bar is needed. For each sentence, the emotion score gets calculated as follows:

1. Sum up all scores for each emotion
2. Divide each sum by number of synsets
3. Return the index of the emotion with the highest score as well as the score

For the visualization, a color index that maps to the color is needed. As we have eight different emotions eight different indices are sufficient. For the color indices see table 4 in chapter 4.6. Visualization.

4.5. Frame Detection

On the FIZ Apollo Website²⁹, we will provide metadata of the analyzed Tweets in the future. However, because of Twitter's Terms&Conditions, we must anonymize the Tweets. Right now, the tool does not provide downloadable files to everyone, but it is available on personal implementation.

Furthermore, we perform Frame Detection (FD) using Framester and store the results in a file. The frames are not used for the visualization itself but will be available in the future for the user to download and work with. The file has the following columns:

Tweet-id, start_index, end_index, lexical_unit, WordNet_Synset, frames

Start_index and *end_index* mark the first and last letter of the selected word within the Tweet. *Lexical_unit* is the LU derived from the selected word. *WordNet_Synset* is the to the LU matching synset received by WordNet. In the *frames* column, all frames which were mapped from WordNet synsets are stored. The mappings are provided by Framester. For the same reasons as discussed in chapter 4.3. we use `grep` instead of a SPARQL query to get the frames.

4.6. Visualization

We want to use an interactive three-dimensional visualization style. Therefore, the WebGL Globe³⁰ is suitable. It is based on the cross-browser 3D library `three.js`³¹. It allows us to visualize data in bars of different heights (representing the scores of the sentiments and emotions in our case) as well as adjust the color of the bars.

The necessary input data is a long array consisting of tuples of four. Each 4-tuple stands for one tweet. The 4-tuple is structured as follows:

latitude, longitude, magnitude, color index

A resulting array could look like this:

[31.8160381, -99.5120986, 0.1238813385, 9, (...), 39.1014537, -84.5124602, 0.144683610875, 5]

²⁹ <http://covid-twitter-stream.fiz-karlsruhe.de/>

³⁰ <https://github.com/dataarts/webgl-globe>

³¹ <https://threejs.org/>

Note that the example above is from the ED results, as the color index of the SA results only ranges from zero to two whereas for ED the indices can go up to 9 (see table 3 and 4, color index column).

We create two globe visualizations, one with the results of the SA, one with the results of the ED. See figures 13 and 14 for examples. For the visualization of SA results, positive sentiments are colored light green whereas negative sentiments are colored red. Neutral scores (score 0) are colored white. However, this appears to be very rare, and as the scores represent the magnitude (height of bars) Tweets with score zero do not have a magnitude. They are visualized as white dots instead of bars. We added a legend so users can interpret the colors on the globe-visualizations.

For the visualization of ED results, eight different colors are needed as DepecheMood returns eight different emotions. To find eight distinguishable colors we used color codes from this website³² (see table 3 and 4).

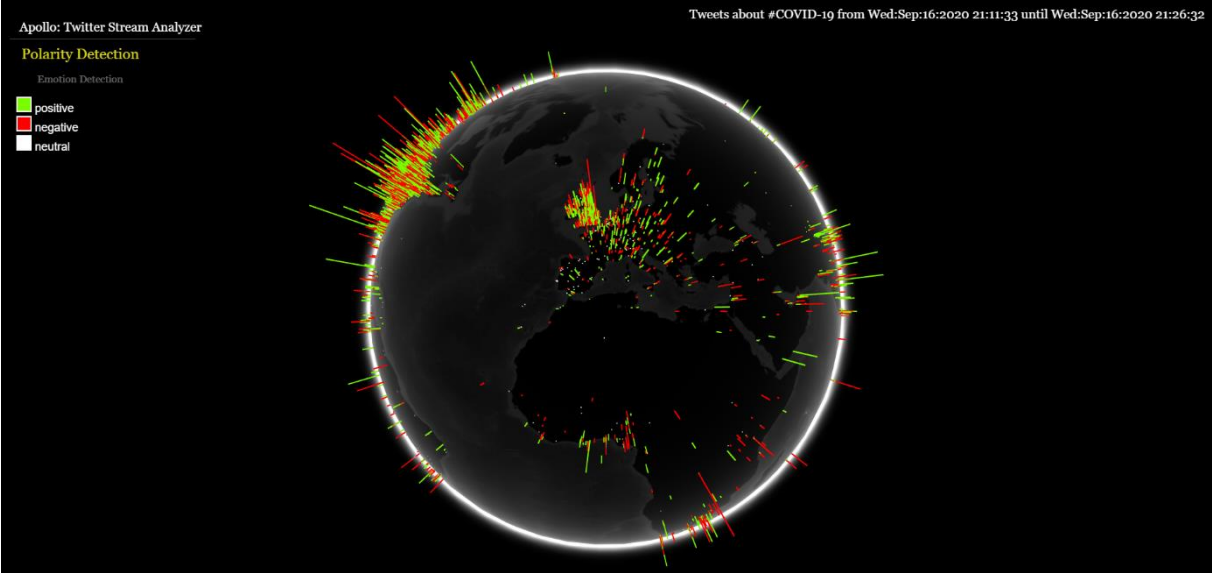


Figure 13 – Sentiment Analysis

³² <https://sashamaps.net/docs/tools/20-colors/>

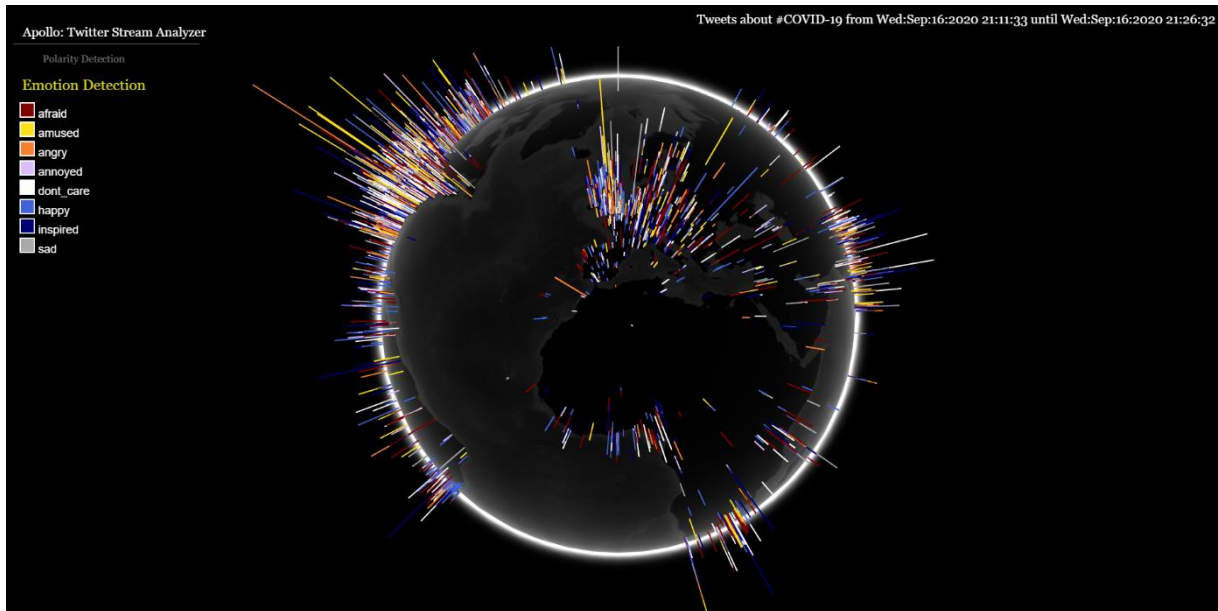


Figure 14 – Emotion Detection

Table 3 – Color codes SA

Sentiment	Color Index	Color code (Hex)
Positive	0	#7CFC00 ■
Neutral	1	#FFFFFF (white)
Negative	2	#FF0000 ■

Table 4 – Color codes ED

Emotion	Color Index	Color code (Hex)
Don't_care	1	#FFFFFF (white)
Afraid	3	#800000 ■
Amused	4	#FFE119 ■
Annoyed	5	#DCBEFF ■
Happy	6	#4363D8 ■
Inspired	7	#000075 ■
Sad	8	#A9A9A9 ■
Angry	9	#F58231 ■

5. Results

The globe visualization demonstrates, that in some countries more Tweets are posted than in others. To find an empirical one-day country distribution, 3318 Tweets over different times during the day have been collected and analyzed for their country code. Most Tweets come from the US and UK, followed by India, Canada, and Australia. The following pie chart (see figure 15) shows the distribution. It is no surprise that more Tweets come from English speaking countries as we analyzed English Tweets only. Furthermore, the US has the highest number of Twitter users worldwide ³³.

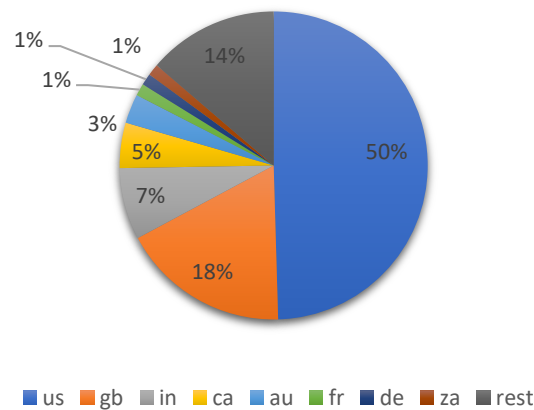


Figure 15 – Distribution of Tweets

Another interesting number is the distribution of sentiments and emotions. Therefore, we counted the sentiments of 5338 Tweets. Of those, 1895 Tweets (~36%) were negative, 2413 Tweets (~45%) were positive, and 1030 Tweets (~19%) were neutral. Considering the negative effects of the pandemic, it is surprising that there are more positive than negative Tweets.

We defined a positive/negative Tweet by a positive/negative average sentiment score calculated by our methodology. For example, one Tweet of a person hoping for a vaccine coming soon was considered as positive as well as a Tweet of a person that was thankful that a relative successfully healed of COVID-19. Tweets about people dying of COVID-19 were considered negative. However, Tweets that had a sarcastic message often got considered positive even though they were negative. To further improve the classification, sarcasm detection could be added, but we will leave that to future work.

³³ <https://www.statista.com/statistics/242606/number-of-active-twitter-users-in-selected-countries/>

The emotions of 4909 Tweets got counted as well and the distribution is shown in the pie chart below (see figure 16). There is no strong bias towards a specific emotion. However, most people were inspired and amused about COVID-19.

We assume that overall opinions and emotions towards COVID-19 greatly differ within society. Most likely it depends on how people got affected by the virus personally.

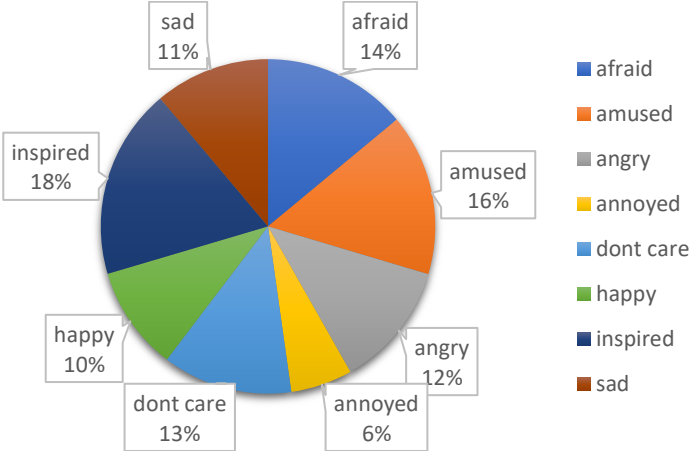


Figure 16 – Distribution of Emotions

Furthermore, as mentioned in chapter 2.1.3. the current algorithm used for WSD (Lesk) is not very accurate (32% accuracy) and thus may result in a false bias of SA and ED. This could happen because the Lesk algorithm may choose a wrong synset with different SentiWordNet scores and different DepecheMood scores than the correct synset. However, most sentences have between five to ten synsets, so if there are enough correct mapped synsets the error gets compensated.

The numbers above were collected over a relatively short time window. We could collect data over time, e.g. one month, or year, and derive and evaluate the overall sentiment towards the pandemic over time. E.g. find out if the overall sentiment was better or worse at the beginning of the pandemic. Or we could connect the SA and ED results with the geotag information and find out if the overall sentiment within countries relates to the number of COVID-19 cases. We will leave that to future work.

The first idea was to have a real-time analysis running, but after the evaluation of the tools, it was clear that without a paid geocoding service it is not possible. There might be other options which we did not find but we will leave that to future work.

6. Conclusion and Future Work

6.1. Summary

In this thesis, after explaining the foundations, we presented a possible solution for analyzing and visualizing Twitter Streams based on trending hashtags. We combined data aggregation on Twitter with NLP (sentiment analysis and emotion detection) and made use of knowledge resources such as SentiWordNet and DepecheMood. The enabling key component to combine WordNet and DepecheMood was Framester. Framester provided mappings between WordNet synsets and DepecheMood emotion annotations. That way we received a positive and a negative score as well as eight different emotions scores for each Tweet. We used the programming language python to manage the tasks above.

Furthermore, we visualized the results on two WebGL Globes, one for the sentiment analysis results and one for the emotion detection results. The color and height of the data points depend on the strength of opinion and emotion derived from the Tweet. The transformation of data points to the visualization was implemented in the programming language JavaScript.

In the results, we evaluated the geo-distribution of the Tweets worldwide and found out that most Tweets come from the US and UK. Furthermore, the amount of positive and negative opinions, as well as the emotions, derived from the Tweets got highlighted. There was no strong bias towards a specific emotion or opinion. However, there were more positive than negative opinionated Tweets which might be because of people joking a lot about COVID-19 or posting sarcastic Tweets.

6.2. Future Work

There are plenty of ways to further optimize the tool. Some of them are:

- Evaluate and implement different possibilities for WSD such as UKB or Babelify
- Increase performance by using a paid geocoding service or combine different free geocoding services
- Add multilingual analysis to have a more representative global coverage of sentiments and emotions
- Add sarcasm detection
- Evaluate and implement different methods for sentiment analysis and emotion detection such as classification with the use of artificial intelligence and deep learning
- Evaluate the results over longer periods, to discover trends over time
- Assess usability of the developed tool, Apollo, also for other topics, than the current corona pandemic
- Improve interpretability of the results and show how this study, might help businesses in strategic decision making and marketing

Declaration about the Thesis

Ich versichere wahrheitsgemäß, die Arbeit selbstständig verfasst, alle benutzten Hilfsmittel vollständig und genau angegeben und alles kenntlich gemacht zu haben, was aus Arbeiten anderer unverändert oder mit Abänderungen entnommen wurde sowie die Satzung des KIT zur Sicherung guter wissenschaftlicher Praxis in der jeweils gültigen Fassung beachtet zu haben.

Karlsruhe, December 30, 2020

Manuel Kaschura

References

- [1] M. M. Mostafa, "More than words: Social networks text mining for consumer brand sentiments: Expert Systems with Applications", pp. 4241–4251, 2013.
- [2] A. Ortigosa, J. M. Martín, and R. M. Carro, "Sentiment analysis in Facebook and its application to e-learning", *Computers in Human Behavior*, vol. 31, pp. 527–541, 2014.
- [3] S. D. H. Evergreen, *Effective data visualization: The right chart for the right data*, 2020.
- [6] Piotr Oleksiak, "Analysing and Processing of geotagged Social Media", 2014.
- [7] Eric Brill and Raymond J. Mooney, "An Overview of Empirical Natural Language Processing", 1, vol. 18, no. 4, p. 13, 1997.
- [9] C. Fellbaum, *WordNet: An electronic lexical database*, 2nd ed. Cambridge, Mass: MIT Press, 1999.
- [10] Y. Shao, C. Hardmeier, and J. Nivre, "Universal Word Segmentation: Implementation and Interpretation", *Transactions of the Association for Computational Linguistics*, vol. 6, pp. 421–435, 2018.
- [11] S. Chea, M. Soeurn, S. Kor, and S. Srun, *Khmer word segmentation with Maximum Matching*.
- [12] N. Bi and N. Taing, "Khmer word segmentation based on Bi-directional Maximal Matching for Plaintext and Microsoft Word document" in *Signal and Information Processing Association Annual Summit and Conference (APSIPA), 2014 Asia-Pacific*, pp. 1–9, 2014.
- [13] P. Tum, *Information retrieval for Khmer documents: challenges and approaches to word segmentation*: California State University, Long Beach, 2007.
- [14] S. Banerjee and T. Pedersen, "An Adapted Lesk Algorithm for Word Sense Disambiguation Using WordNet" in *Computational Linguistics and Intelligent Text Processing*, Berlin, Heidelberg, pp. 136–145, 2002.
- [15] S. Mccroy, "Using Multiple Knowledge Sources for Word Sense Discrimination", *Computational Linguistics*, vol. 18, pp. 1–30, 1992.
- [16] H. T. Ng and H. B. Lee, "Integrating multiple knowledge sources to disambiguate word sense" in *Proceedings of the 34th annual meeting on Association for Computational Linguistics -*, Santa Cruz, California, pp. 40–47, 1996.
- [17] M. Lesk, "Automatic Sense Disambiguation Using Machine Readable Dictionaries: How to Tell a Pine Cone from an Ice Cream Cone" in *Proceedings of the 5th Annual International Conference on Systems Documentation*, pp. 24–26, 1986.
- [22] A. Moro, A. Raganato, and R. Navigli, "Entity Linking meets Word Sense Disambiguation: a Unified Approach", *Transactions of the Association for Computational Linguistics*, vol. 2, pp. 231–244, 2014.
- [23] E. Agirre and A. Soroa, "Personalizing PageRank for Word Sense Disambiguation" in *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, 2009.
- [24] C. Fillmore, "The case for case", 1967.
- [25] H. C. Boas and R. Dux, "From the past into the present: From case frames to semantic frames", *Linguistics Vanguard*, vol. 3, no. 1, 2017.
- [26] Charles J. Fillmore, "Scenes-and-frames semantics" in 1977.
- [27] M. Petruck, "Frame Semantics", 2003.
- [28] Charles J. Fillmore and B. T. S. Atkins, "Toward a Frame-Based Lexicon: The semantics of Risk and its Neighbors" in 2015.

- [30] T. Berners-Lee, J. Hendler, and O. Lassila, "The Semantic Web: A New Form of Web Content That is Meaningful to Computers Will Unleash a Revolution of New Possibilities", *ScientificAmerican.com*, 2001.
- [31] L. Yu, *A Developers Guide to the Semantic Web*. Berlin, Heidelberg: Springer-Verlag Berlin Heidelberg, 2011.
- [34] K. Balog, *Entity-Oriented Search*. Cham: Springer International Publishing, 2018.
- [37] J. Lehmann *et al.*, "DBpedia – A large-scale, multilingual knowledge base extracted from Wikipedia", *Semantic Web*, vol. 6, no. 2, pp. 167–195, 2015.
- [38] P. Mendes, M. Jakob, and C. Bizer, "DBpedia: A Multilingual Cross-Domain Knowledge Base", *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, 2012.
- [39] R. Sofronova, "Fine-grained Type Prediction of Entities using Knowledge Graph Embeddings", Karlsruhe, 2019.
- [43] P. Cimiano, C. Chiarcos, J. P. McCrae, and J. Gracia, "Linguistic Linked Open Data Cloud" in *Linguistic Linked Data: Representation, Generation and Applications*: Springer International Publishing, pp. 29–41, 2020.
- [44] A. Gangemi, M. Alam, L. Asprino, V. Presutti, and D. R. Recupero, "Framester: A Wide Coverage Linguistic Linked Data Hub" in *Lecture Notes in Computer Science, Knowledge Engineering and Knowledge Management*, E. Blomqvist, P. Ciancarini, F. Poggi, and F. Vitali, Eds., Cham: Springer International Publishing, pp. 239–254, 2016.
- [45] B. Pang and L. Lee, "Opinion Mining and Sentiment Analysis", *FNT in Information Retrieval*, vol. 2, 1–2, pp. 1–135, 2008.
- [46] B. Liu, *Sentiment analysis: Mining opinions, sentiments, and emotions*. Cambridge: Cambridge University Press, 2015.
- [47] A. Esuli and F. Sebastiani, "SENTIWORDNET: A Publicly Available Lexical Resource" in *Proceedings of 5th International Conference on Language Resources and Evaluation (LREC)*, pp. 417–422, 2006.
- [48] D. Reforgiato Recupero, V. Presutti, S. Consoli, A. Gangemi, and A. G. Nuzzolese, "Sentilo: Frame-Based Sentiment Analysis", *Cogn Comput*, vol. 7, no. 2, pp. 211–225, 2015.
- [49] E. Chen, K. Lerman, and E. Ferrara, "COVID-19: The First Public Coronavirus Twitter Dataset", 2020.
- [51] D. Dimitrov *et al.*, *TweetsCOVID19 - A Knowledge Base of Semantically Annotated Tweets about the COVID-19 Pandemic*.
- [53] I. Li, Y. Li, T. Li, S. Alvarez-Napagao, and D. Garcia, "What are We Depressed about When We Talk about COVID19: Mental Health Analysis on Tweets Using Natural Language Processing", Apr. 2020.
- [54] T. Klomklao, P. Ratanarungrong, and S. Phithakitnukoon, "Tweets of the nation" in *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct : September 12- 16, 2016, Heidelberg, Germany, Heidelberg Germany*, pp. 1349–1357, 2016.
- [55] E. Duarte, P. Bordonhos, P. Dias, and B. S. Santos, "Living Globe: Tridimensional Interactive Visualization of World Demographic Data" in *Human Interface and the Management of Information: Information, Design and Interaction*, pp. 14–24, 2016.