Università degli Studi di Salerno

Dipartimento di Ingegneria dell'Informazione ed Elettrica
e Matematica Applicata

Course of

BIG DATA E TECNOLOGIE SEMANTICHE

finNSEMA, a tool for financial market analysis using semantic representation of financial news

Group finnish

Marco Carpentiero, Gerardo Corbisiero, Ilaria Gigi, Giuseppe Seccia, Antonio Vicinanza

Academic year 2019/2020

Contents

1 Introduction **1**

2 System description **4**

2.1 Technologies 4

2.2 System architecture 4

2.2.1 Functional architecture 5

2.2.2 Software architecture 5

2.3 System ad-hoc ontology design 7

2.4 Modules description and functionalities analysis 9

2.4.1 Data extraction pipeline 9

2.4.2 Data storage 12

2.4.3 Data visualization and inference 13

2.5 GUI Description 17

3 Conclusions **20**

References **23**

List of figure **23**

List of tables **24**

It is well enough that people of the nation do not understand our banking and monetary system, for if they did, I believe there would be a revolution before tomorrow morning.

– Henry Ford

1

Introduction

It is claimed that more and more companies, today, are looking for a profitable and complete exploitation of cutting-edge techniques in their daily products and services, such as machine learning, big data analytics and semantic web, which have had a capillary diffusion in recent years. Many enterprise entities have been charmed by the trend mentioned above, but it is clear that one of the most engaged is finance. The reason is crystal clear.

The financial services providers are investing billions [8] in all those new technologies that may speed up the analysis of financial markets or national economies data. In this sight, there have already being developed systems which try to automate a part of the financial industry workflow. An area of algorithmic dominance that often goes unnoticed is in the stock market. These trading algorithms are reshaping the way trading is done on Wall Street. Investors are using algorithms designed for trading to bring greater efficiency to financial markets and this leads to the so-called "algorithmic trading". In fact, by eliminating the emotional (thus potentially biased) choice of a human trader, an autonomous system, like the one aforementioned, could increase the investments revenue. Just to get in touch with this new financial approach, it should be considered the following striking data. In 2006, at the London Stock Exchange, a financial colossus, over 40% of all orders were committed by algorithmic traders [14]. In the US, about 70 percent of overall trading volume is generated through algorithmic trading. A recent report estimated that the world market for algorithmic trading will grow by 10.3% CARG from 2016 to 2020 [5].

The main issue in this strategy is that it has been thought up to be based on information mostly extracted from purely financial data, which means: trade values, stock value trend, economic gain and losses, percentage variations of a stock index and so on. Whereas it should

not go unnoticed that, in the current pervasive information era, these data are scattered and highly correlated with the mindset of the financial audience (from the smallest investor to the most eligible company) by glimpsing the plethora of economics information publicly available about a specific topic, roughly speaking the news and "gossips". In fact, consider a news reporting that a quoted company is being charged with accounting fraud, there is no doubt that some news agency will write about it, triggering a chain reaction in the investors. Hence, some of them will decide to sell their stocks, others will wait for the value of the company to go down to reinvest in it, hoping in a future payback, still others will just stare at their portfolios; but the financial machine has unavoidably turned on, producing the effects that, maybe, an (autonomous) algorithmic trader could catch to make efficient market moves.

So, what if you could somehow anticipate this phenomenon by taking a wiser look at the news and information about economics? It would mean to be a step ahead the market movements and, maybe, to be able to optimize trading strategies thanks to a more clever approach, based on "the cause" and no more on "the effect", that is, generally, the key to the success of rich and famous investors, naturally gifted with this talent.

There comes the intuition to develop "finNSEMA" (financial News SEMantic Analyzer). The tool drives itself in a free action space since no other tool with similar features is currently available.

finNSEMA collects financial news from the most valuable news agencies and analyzes them to extract trusted data and trusted entities of interest to dynamically populate a custom semantic knowledge base, ad-hoc designed, to maximally exploit news related financial information. Thus, a trader using finNSEMA can invest having a thorough consciousness of the up-to-the-minute state of art of the financial world. That is because its current status is well structured and easily accessible since news contents and related data are wrapped in a fixed structure and linked each other with powerful semantic meaning. The collected linked data can be navigated so that smart and otherwise hidden associations between financial entities can be found thanks to the semantic analysis. This could lead to successful trading moves based on an unpleasant amount of news unfeasible to be read, comprehend, deeply understand, contextualize and more. finNSEMA does all these jobs as its primary mission, thanks to the technologies already mentioned above (Machine learning, Named-entity recognition, Web scraping, etc) and provides customized results in an attractive and user-friendly framework.

References

Link to GitHub project: `https://github.com/gseccia/FinancialNewsSemanticScraper`

2

System description

2.1 Technologies

finNSEMA is fully written in Python and incorporates the following technologies.

Name	Reference	Description
Protégé	[13]	Graphical tool for interactive ontology building
Tarsier	[3]	3d viewer for RDF knowledge bases
NLTK	[10]	Platform for building Python programs to work with human language data
ParallelDots	[11]	API to identify individuals, companies, places, organization, cities and other various type of entities
Apache Jena Fuseki	[4]	SPARQL server
Selenium	[1]	Framework for browser automation
PyQt	[9]	Python bindings for the Qt cross-platform C++ framework
Keras	[7]	Deep Learning library for Python

Table 2.1: finNSEMA technologies.

2.2 System architecture

finNSEMA is built around a core back-end block, that manages data storage, retrieval and manipulation, connected to a front-end block assigned to data visualization and inspection. The latter activity is the main user task, who is comfortably separated from the technical core of the product and is bounded by a GUI for the setup of the system and a web GUI to actually visualize data in a semantic oriented environment.

2.2.1 Functional architecture

Figure 2.1 is a compendium of the finNSEMA tool architecture, from a functional point of view. It highlights the main operations performed by the specific components that have been integrated. Some of them have been developed to accomplish very specialized tasks, while others are available and have only been integrated. For further details, please refer to section 2.4

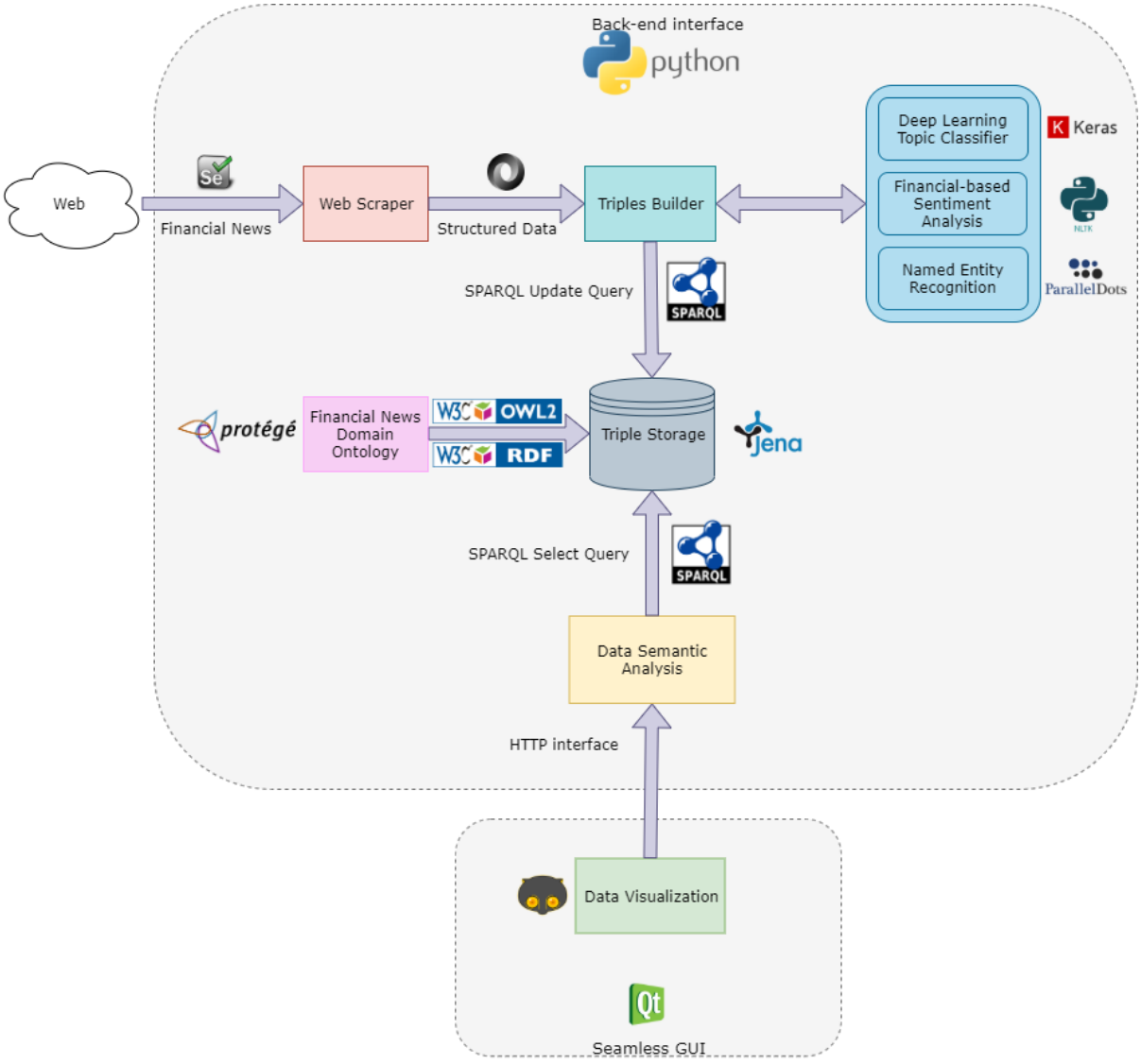

Figure 2.1: finNSEMA functional architecture. All the technologies exploited by finNSEMA are also shown.

2.2.2 Software architecture

finNSEMA is an object-oriented software, written in Python programming language. Figure 2.2 shows the UML class diagram, containing the relationship existing between the software

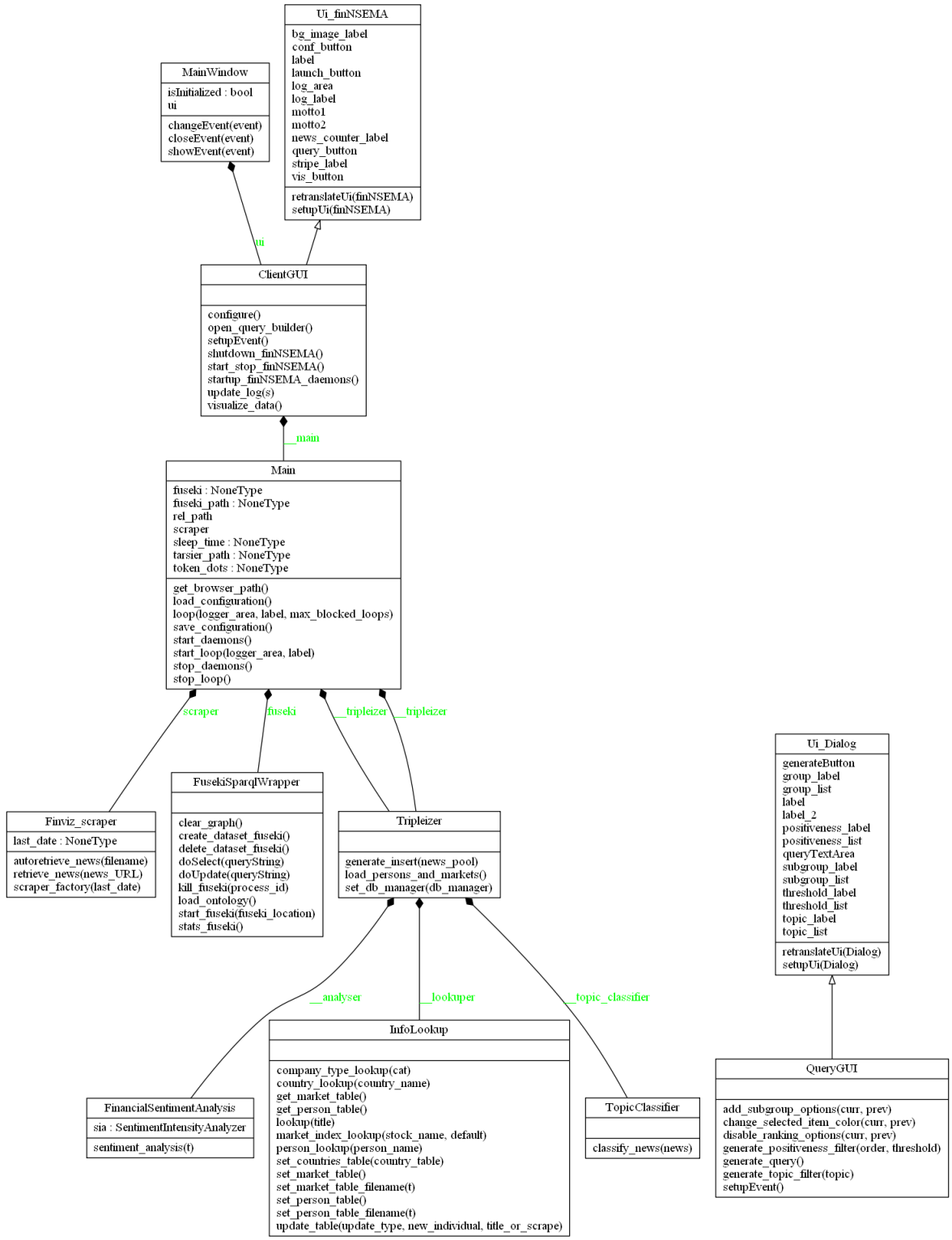

Figure 2.2: finNSEMA class diagram

modules. The starting point of it is the user interface, implemented in ClientGUI class and nestled in a MainWindow class from PyQt5 library, whereas the center of the system is the Main class, which has the "crawling" role and handles all the other functionalities (Finviz_scraper represents the "scraping", FusekiSparqlWrapper is the high-level interface of the Database, Tripleizer takes care of constructing sparql triples from raw data, etc.)

2.3 System ad-hoc ontology design

Using semantic technologies in the financial domain is still a young activity, so finding a supportive domain ontology to forge finNSEMA knowledge base is not easy. The first solution you can bump into is FIBO [2]. It is a remarkable and large project but it turned out to be too detailed and actually far from financial news context. A more coherent arrangement with respect to finSEMA aim, was found in [8]. This paper submits an ontology partially covering the financial domain, which excluded many important concepts of the field such as the relationship between news, companies, people. It is clear that this is unsatisfactory.
No other remarkable state-of-the-art solutions were discovered, therefore there have been the need to design an ad-hoc ontology to appropriately fit the contents of financial news that could be of interest for an expert.
A comprehensive view of the ontology is in figure 2.3, in which the whole set of classes, object properties and data properties, is shown. The ontology exposes the following concepts, in the form of OWL classes:

- **News**: main class of the ontology, representing the individual news analyzed by the finNSEMA system;
- **Organization**: generic representation of a company involved in financial news. It is specialized in a set of typologies that reflect the ones defined by financial news publishers. To establish them, it has been made a mapping between Bloomberg classification and TRBC standard classification [15], as a compromise between enough conciseness of the first and exhaustiveness of the second;
- **Countries**: country concept representation. finNSEMA uses countries definitions borrowed by a geographic ontology of W3C property;
- **Person**: person concept of foaf ontology, borrowed from it;

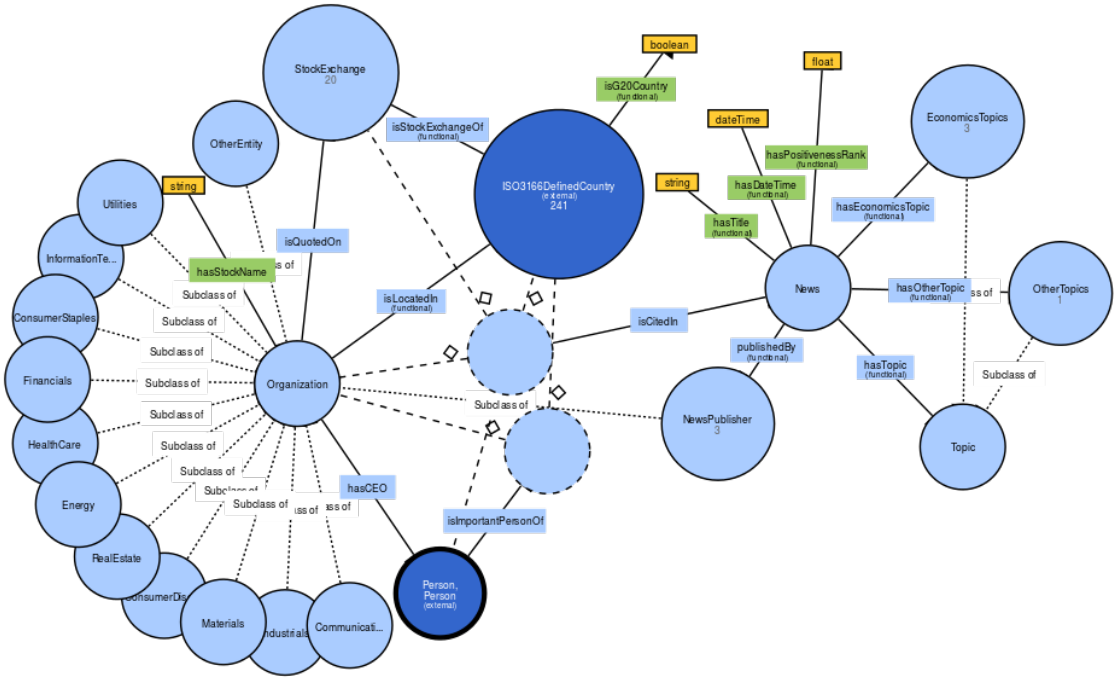

Figure 2.3: finNSEMA ontology graph representation.

- **StockExchange**: stock exchange conceptualization;
- **Topic**: broad indication of the topic of the analyzed news, as economics news or not.

This group of classes takes advantage of the following object properties:

- **isCitedIn**: indicates the citation of an entity of the ontology in a news;
- **hasTopic**: indicates the typology of topic of a given news. It is divided into the subproperties hasEconomicsTopic and hasOtherTopic, with obvious meaning;
- **publishedBy**: indicates the news publisher of a given news;
- **hasCeo**: indicates the chairman or chief executive officer of an organization, if available;
- **isLocatedIn**: indicates the legal site of an organization;
- **isQuotedOn**: indicates on which national market index a given organization is quoted;
- **isStockExchangeOf**: indicates the relation between a national market index and the corresponding Country entity;
- **isImportantPersonOf**: indicates a meaningful relation between a Person and a Country or Organization.

The strength point of this knowledge base is the dense and very significant connection, as clear from figure 2.3, between its concepts. It must be contended that all the concepts are connected to the main concept News via isCitedIn and hasTopic properties. This implies that, from a given News, it is possible to collect all the financial domain information related to it and explore traversing a secondary level of indirection, due to the other properties of the found information, many more useful data.
As an example, consider the following news title: "Tesla wants to invest 4 billion dollars to produce rocket cars in Kazakhstan". finNSEMA analyzes the contents and finds some cited concepts in the news, such as Tesla, an Organization, and Kazakhstan, a Country. Exploring the ontology graph from the found classes individuals it is possible to realize that Tesla is an Industrial company, quoted on Nasdaq stock exchange, located in the USA, Elon Musk is its CEO and so on. All these data are free, just by recognizing some concepts in the news allowing a broader insight of its financial meaning.
The ontology was developed using Protégé [13] and the OWL2 syntax, then serialized in RDF/XML to be managed by the triple storage Apache Jena Fuseki.

2.4 Modules description and functionalities analysis

The following sections give some more details on the building blocks of figure 2.1 and their role in the data processing workflow.

2.4.1 Data extraction pipeline

The data used by the application are retrieved by the Web scraper module that connects to the online publisher `https://finviz.com/news.ashx`. This website contains all the primary financial news posted by the most influent news agencies: Reuters, Bloomberg, BBC, The New York Times etc.

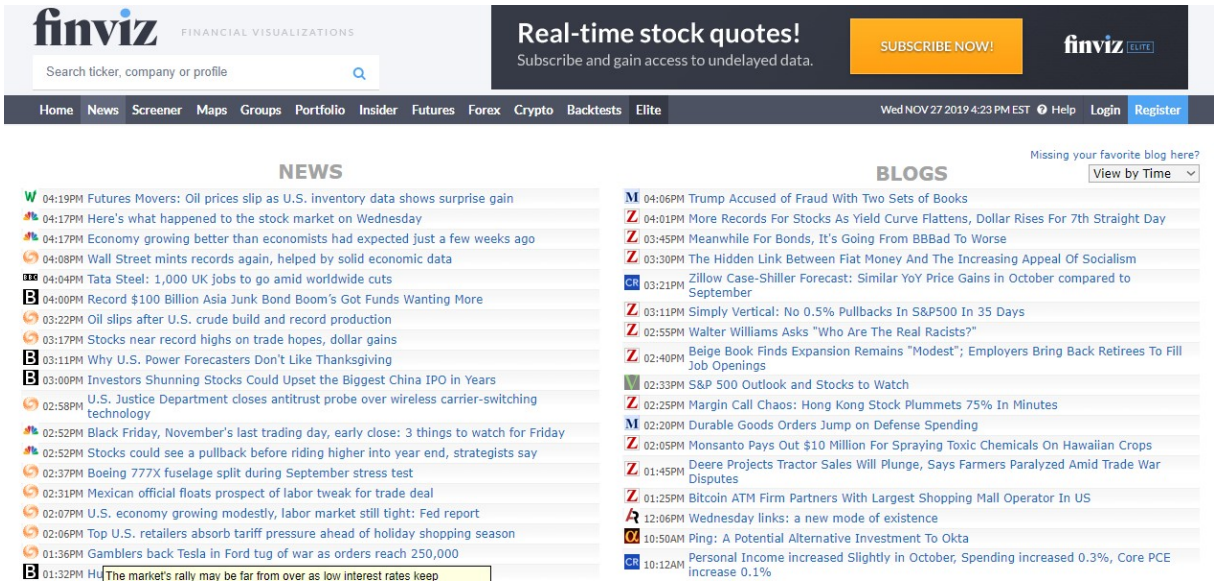

Figure 2.4: finviz.com news portal.

The Web scraper module filters the contents of the website, selecting only news published by Reuters and Bloomberg, because these sources are the most trustworthy and provided of useful and valuable information.

Basically, the scraper module exploits the visited web pages structure to build a JSON object containing only the chosen set of data about the financial entities cited into the news: news title, company name, company type, stock exchange index, location and so on. The primary goal of this operation is to generate a structured set of data, starting from a non-structured data format exposed as text in web pages.

Web scraper module is based on the Selenium library [1] and, as touched upon, checks the visited page HTML tags to "steal" useful data and follows hyperlinks associated to the financial entities of interest. For each of these links it applies another level of scraping to ultimately fill the JSON object.

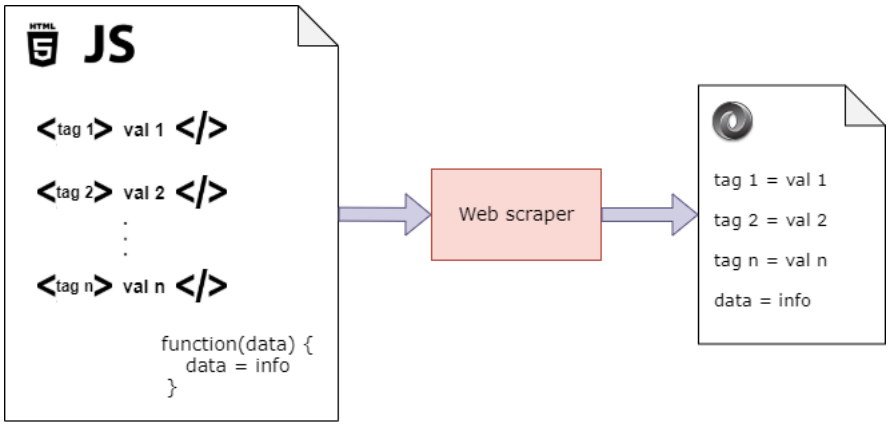

Figure 2.5: Web scraper activity.

At the end of this procedure the structured data is fed to the Triples Builder module of the architecture. On the assumption that the ontology has a fixed structure, this means that it does not change along execution time, and that is fully known by Triples Builder, the scraped data can be easily transformed in a set of Turtle triples compliant with the ontology graph. To accomplish this task, Triples Builder scans the JSON object and creates, with the labelled data in it, the triples that have to be inserted in the knowledge base.
Some triples only includes "scraped" data while others must be computed with the aid of other modules, that provides concepts which cannot be retrieved by the scraper alone. These procedures are:

- **Named Entity recognition**: since the news title is a non-structured piece of information, the scraper is not able to find any useful data in it. However, this text may still rich of important financial properties, for instance companies or persons mentioned in it. So, a way to extract this knowledge is absolutely necessary. The solution is the usage of a Named Entity Recognizer (NER). After several researches and alternatives taken into account, this module uses the ParallelDots API [11] as NER because it has provided a good confidence interval within the financial news domain. This process is particularly important because it links the entities mentioned in the news title with the corresponding content scraped in the web page through a set of triples, establishing a consistent knowledge base;

- **News Topic Classification**: in order to partition the space of News, the module approaches the problem with a machine learning based news topic classification. A deep neural network, built with Keras framework [7], was trained on labelled news titles in order to predict for each news title the belonging to one of the following classes: CompaniesEconomy, Markets&Goods, NationalEconomy or OtherTopic. This choice leads to the opportunity of filtering news to focus the attention on a smaller set of data;

- **Sentiment Analysis**: a suggestion for the user about the positiveness of the content of the news, based solely on the title text, is generated with a sentiment analysis that returns a numeric value in $[-1, 1]$. The module provides this feature using NLTK library [10] tuned with a specific financial lexicon provided in [6]. This result is then added to a triple that encodes the fundamental property on which the user market analysis can be carried on.

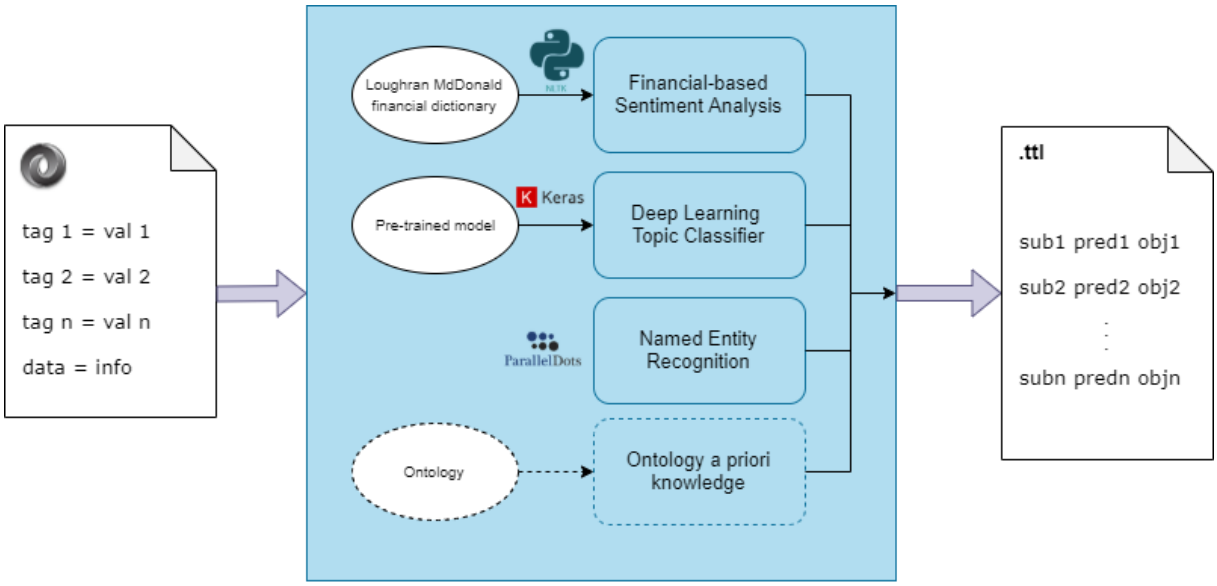

Figure 2.6: Triples Builder activity.

2.4.2 Data storage

As mentioned in section 2.3, finNSEMA stores its data in one of the most used triples storage, that is Apache Jena Fuseki [4]. The set of triples extracted in the pipeline of section 2.4.1 are inserted in the database using the SPARQL Update syntax. The update operation is wrapped using the SPARQL over HTTP interface offered by Fuseki, without the need of any actual database manager to interface it, lightening finNSEMA back-end core. Data storage is chosen persistent.

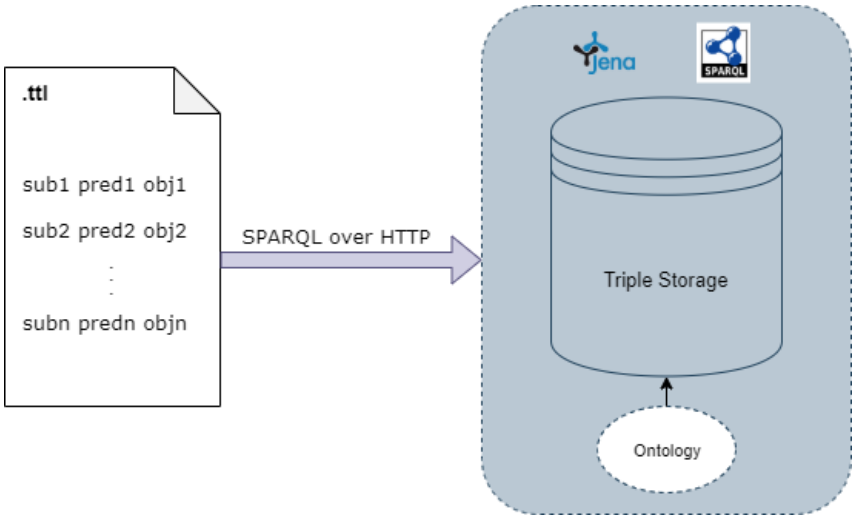

Figure 2.7: finNSEMA storage organization.

2.4.3 Data visualization and inference

finNSEMA leans on Tarsier [3] for data visualization and inference. Tarsier is a 3D viewer for RDF knowledge bases, based on graph construction using HTTP requests embedding SPARQL queries to SPARQL endpoints and on parsing the results as JSON. Unfortunately, it is only an experimental project not fully developed yet and needed some fine tuning to match finNSEMA architecture.

Tarsier visualization works by creating an inner RDF graph with the results of a SPARQL Construct query. To this knowledge base, different filters, both available in the user interface both obtainable with custom SPARQL queries, can be applied in order to build "semantic planes". A semantic plane is a plane in a 3D space on which a filter result lies, still preserving possible relationships (in the form of arcs) with other semantic planes entities.

In finNSEMA, Tarsier is connected to the Apache Jena Fuseki SPARQL endpoint to retrieve query results. Its basic knowledge base graph is built through a Construct query that collects all the information around the scraped news, in finNSEMA ontology terms (classes, dataProperties, objectProperties, etc); whereas, the filters can be made through buttons on Tarsier interface or custom queries. Some possible queries of domain interest are provided by finNSEMA user interface and some others are those available in Tarsier configuration file, but many more can be invented!

In order to make a proper use of Tarsier view, the user must first construct the graph, then apply a filter (or more than one) and check the results in the dedicated plot area.

finNSEMA query system allows the user to produce a large number of filtering queries combining all the possible query fields choices available in the GUI. The purpose of the queries is to isolate the triples associated to a main concept (Companies, Stock Markets etc.) that is related to a financial news, via direct citation or data scraping. More than selecting the concept, the user can add (optionally) filtering levels in order to select information involved only in news with a positiveness level lower/greater than a threshold and in news of a given topic.

To make inference on data, a trader can apply one or more filters and visually inspect the results in the different semantic planes. Each plane embodies the the results of a filter. The trader can, thus, apply financial operations on the basis of the information he can inspect in the semantic planes chosen. Having full control on the inference process just explained, finNSEMA ensures security and safety for the trader capitals. Also, this "man in the loop"

characteristic gives more confidence to the trader that keeps full control on his financial operations, differently from what may happen with algorithmic traders.

Usage example

To show the tool capabilities, an use case is provided in this section. The starting knowledge base is shown in the below image from Tarsier plot.

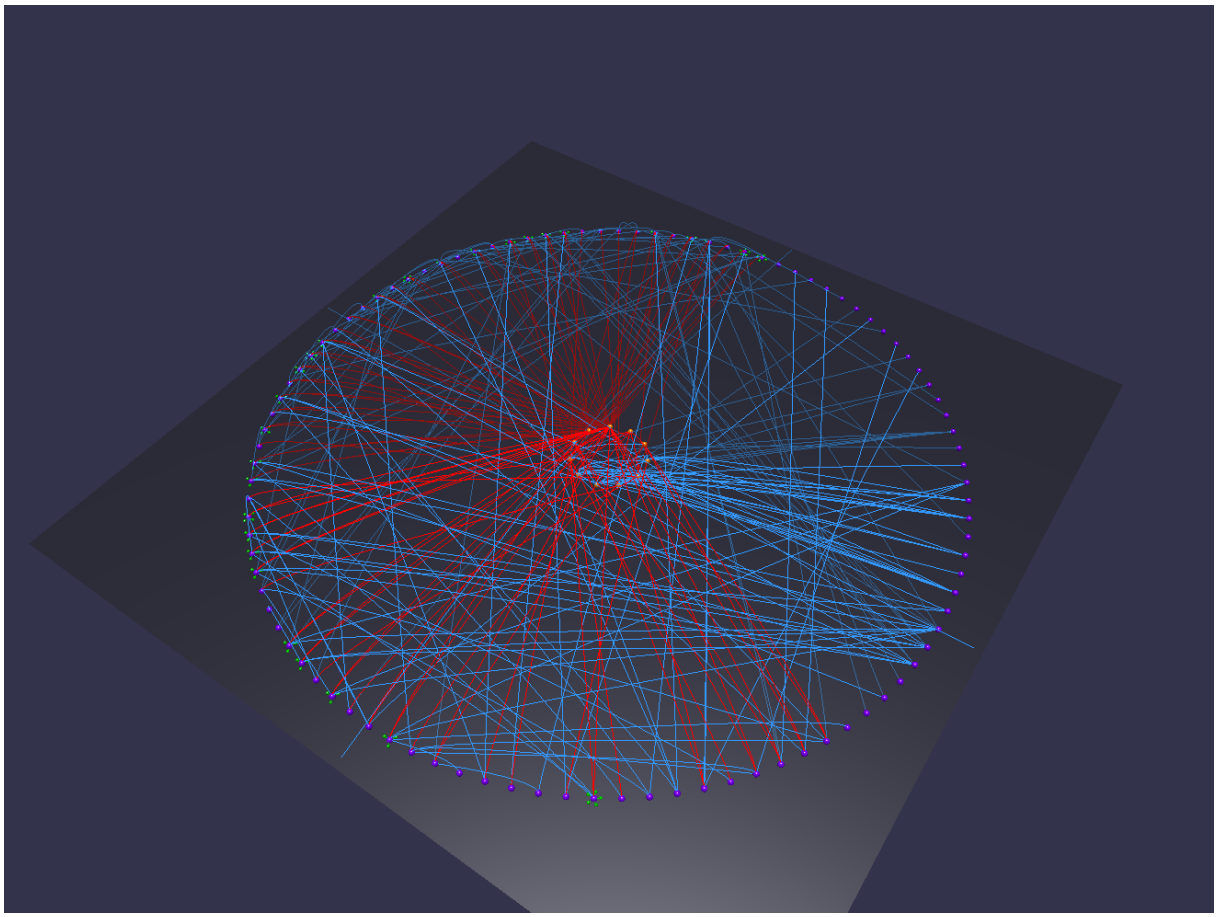

Figure 2.8: Knowledge base extraction.

In a second step only concepts related to positive news are retained, using the predefined filtering query available in Tarsier.

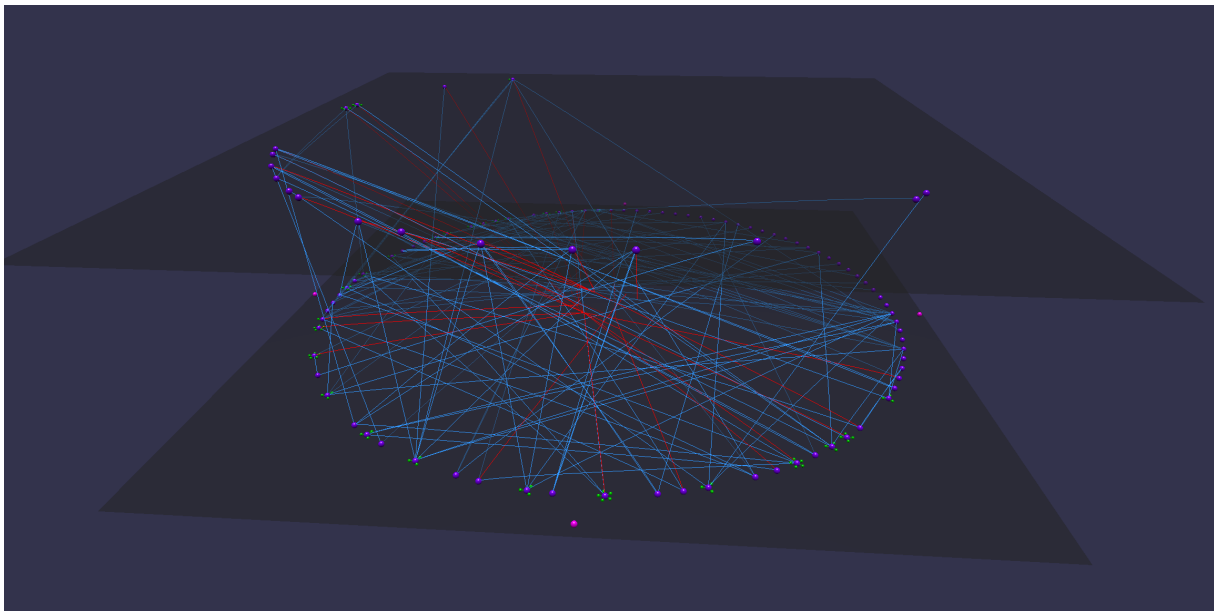

Figure 2.9: The upper plane contains filtered data according to "concepts related to positive news".

Thus, any filters can be applied to the current highest plane to show other semantic planes. For example, using a filter about Companies of "Consumer Discretionary" industry sector the result is in the following figure:

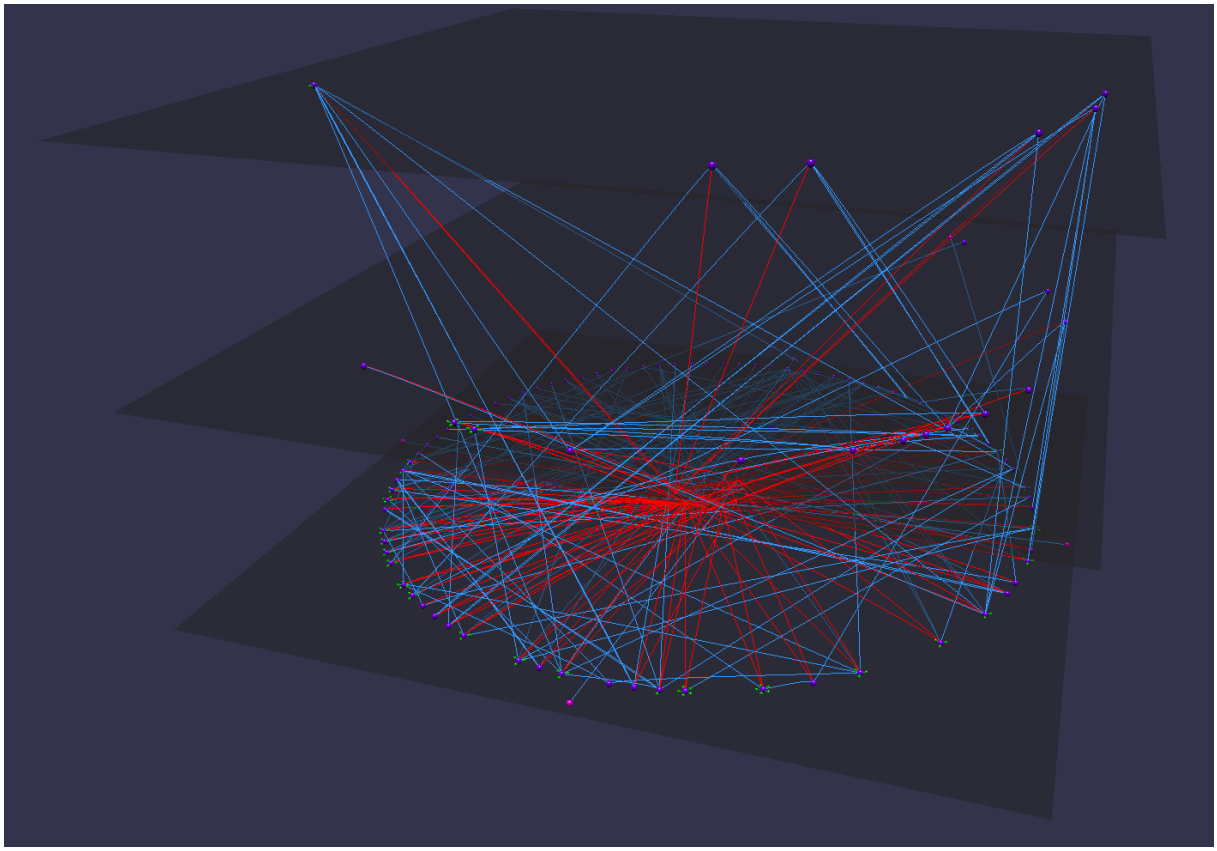

Figure 2.10: The upper plane contains filtered data by Companies

To strengthen, in this example, the semantic meaning, it is be possible to create another plane searching for Companies associated to both positive news and negative news, so that Companies associated to only positive news are highlighted.

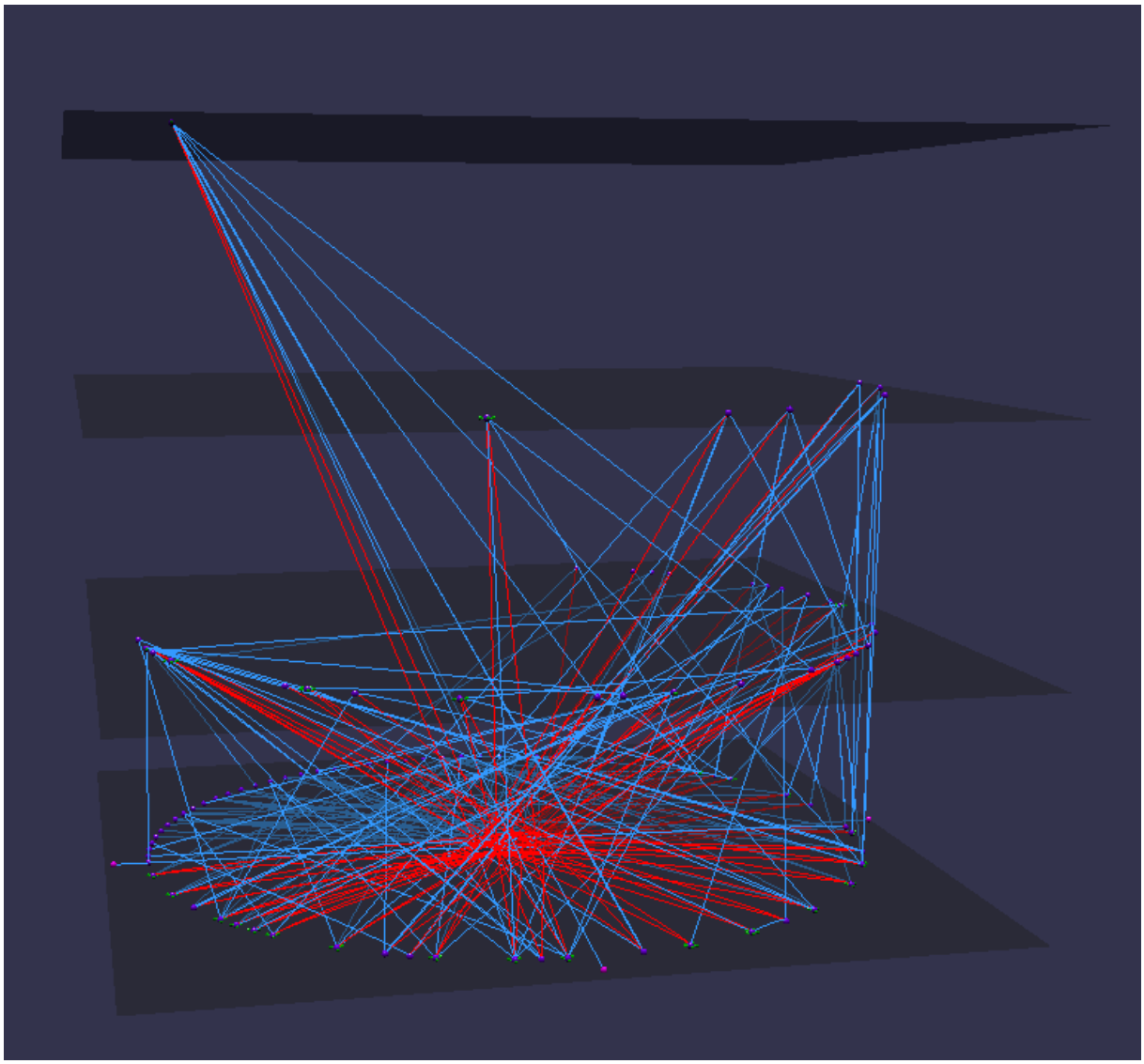

Figure 2.11: The upper plane contains filtered data by "negative concepts"

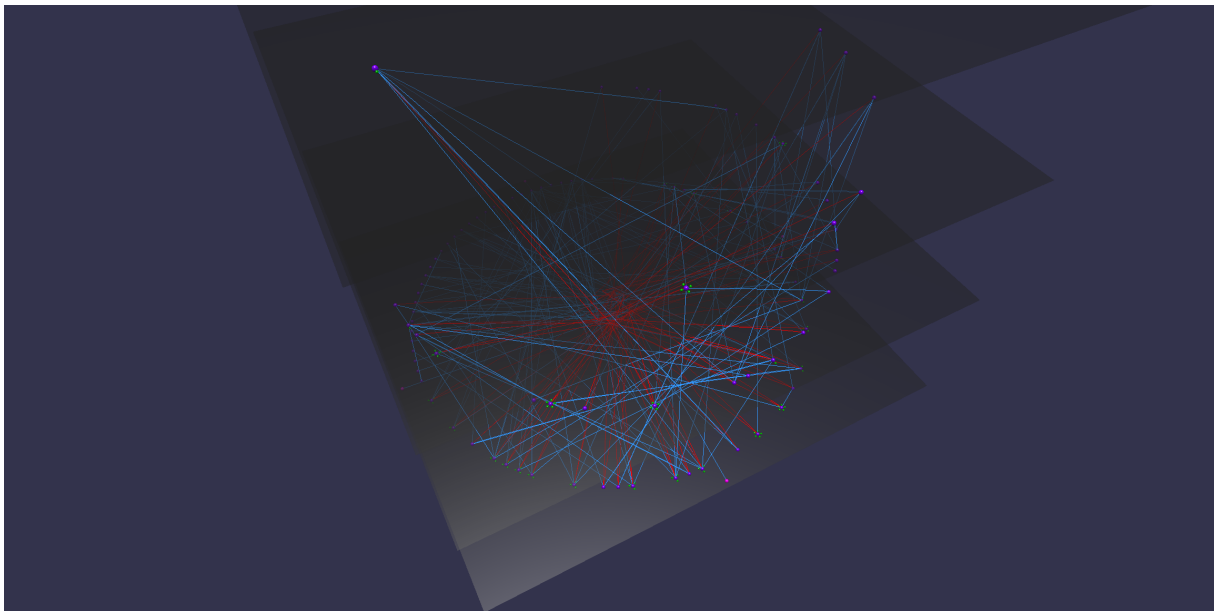

Figure 2.12: Another perspective view

In conclusion, the analysis proves that you should invest in Companies come out in the third layer (positive-news related Companies) and not in those in the fourth layer (negative-news related Companies).

2.5 GUI Description

The GUI of finNSEMA has been designed to allow the user to use the front-end functionalities in a friendly way, completely hiding the complexity of the underneath back-end processes, all encapsulated in a captivating look. It is provided of the following buttons:

- **Settings Button**: It opens a file browser to select the own configuration file; of course, it must be well-formatted or the system cannot work properly. The file must contain a JSON including the following parameters:

```
{
    "browser_path": "path/to/chromedriver/",
    "first_start": false/true,
    "fuseki_path": "path/to/fuseki/jar/",
    "news_update": in seconds,
    "tarsier_path": "path/to/tarsier/",
    "token_paralleldots": provided by the license
```

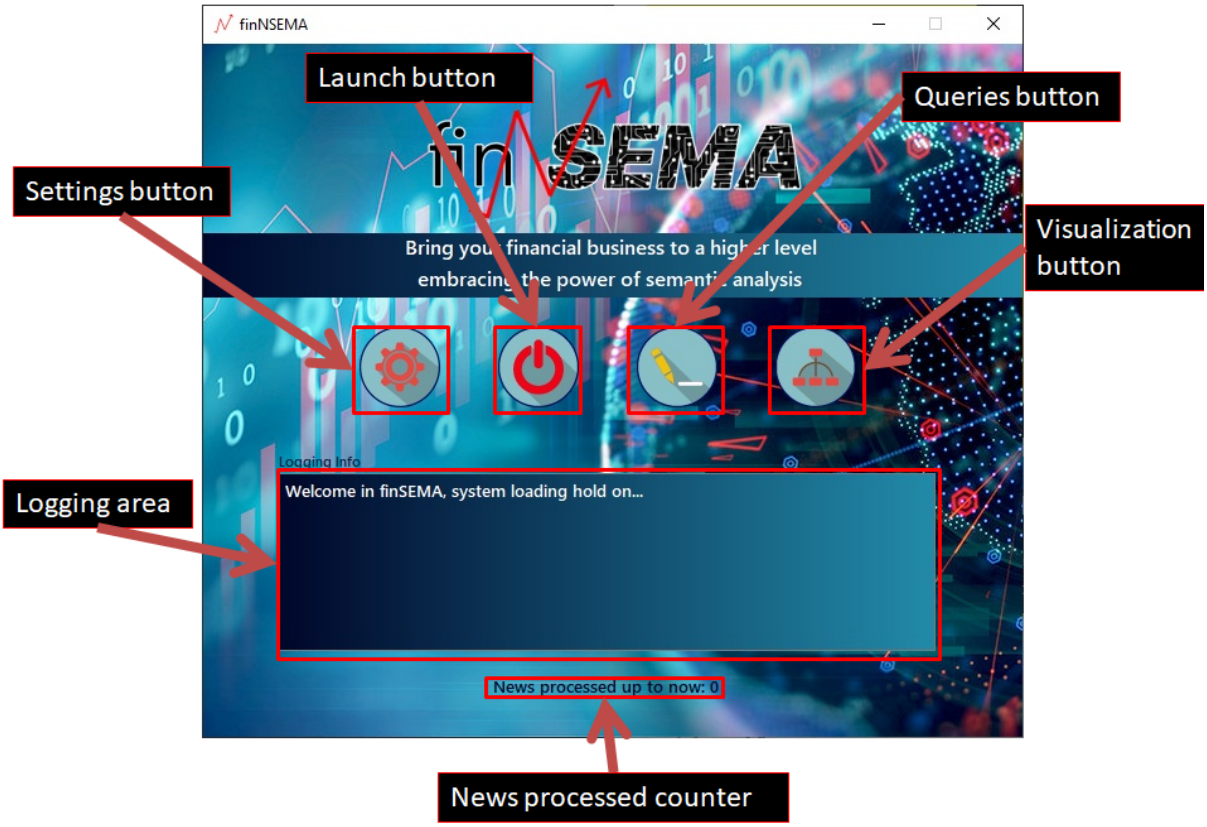

Figure 2.13: finNSEMA GUI

```
    }
```

- **Launch Button**: It allows to launch and stop the data extraction process;
- **Query button**: It opens another window that allows to build specialized queries;
- **Visualization button**: It opens the Tarsier page in the browser for data visualization;
- **Logging Area**: It contains information messages for the user coming from the underlying software;
- **News processed counter**: It counts the news processed up to now.

When the query button is clicked, the following window is opened.

The user can interactively build a query selecting a parameter from the lists available. They allow to specify the query attributes and semantic filters:

- **Group**: indicates the main subject of the query, for example persons, companies and more;

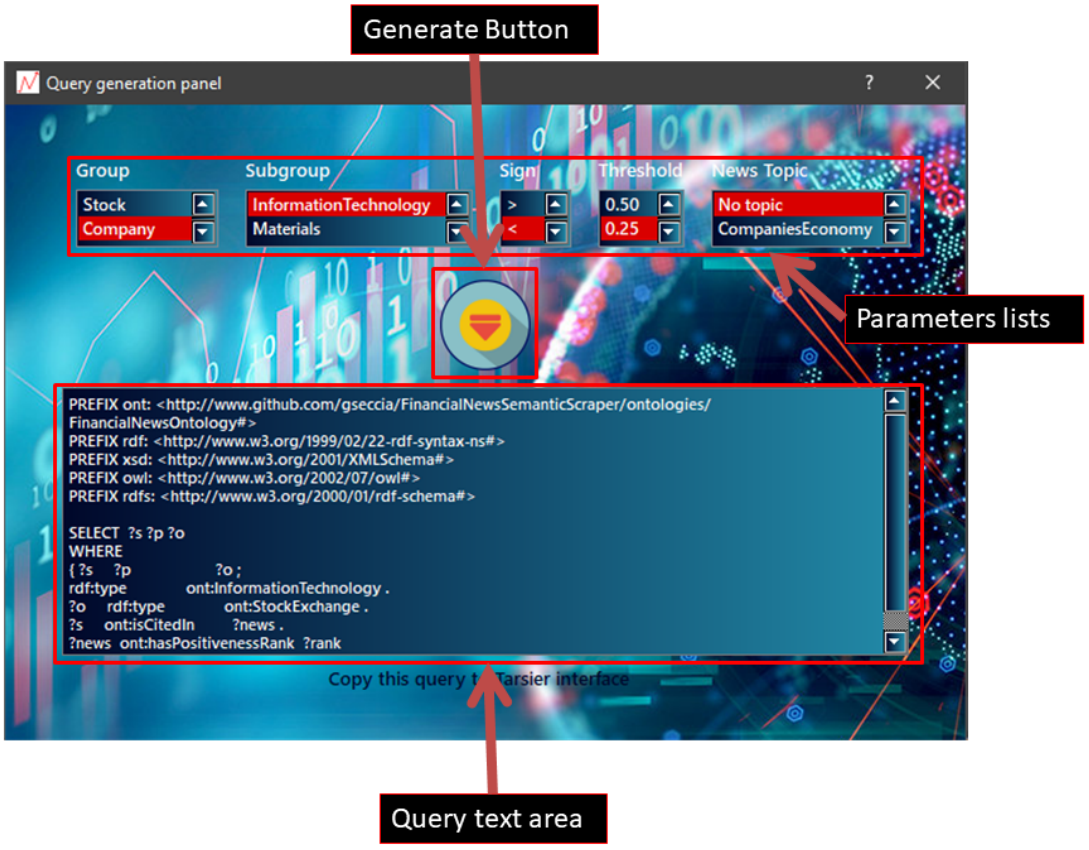

Figure 2.14: Query generation panel GUI.

- **Subgroup**: indicates a subdivision of the Group entity, depending on the selected group;
- **Sign**: indicates if to set a news positiveness filter, choosing the relational operator ($>$, $<$);
- **Threshold**: indicates the positiveness threshold to use if the Sign filter is set;
- **News Topic**: indicates the news topic.

By clicking "generate" button, on the center of the window, the query appears in the below text area. If some of the parameters are missing, the user is properly warned.
Thus, the created query can be copy-paste in the apposite query area of Tarsier, to filter the data according to the user needs.

3

Conclusions

While it is true that there is a lack of tools for the semantic knowledge construction in finance domain, as explained at the beginning of this document, also finNSEMA manages to fill this gap for financial news analysis, in order to help long term investors to keep a close watch on financial markets, succeeding in building robustness to negative events (in sight of a feasible "black swan"[12]) and the capability to exploit positive events.
Anyway, some improvements could be implemented to enhance the architecture:

- A physical separation between the "back-end" part and the "front-end" part of the system, which means the data extrapolation/storage and the data visualization/analysis. This would allow to get a reliable centralized server-sided endpoint (i.e: a Web Service) accessible through the Internet to more than one client per time. To achieve this goal, therefore, there is no little effort since the software architecture must be reorganised and, most of all, data security and consistency must be insured, allegedly with a new module from scratch since Apache Jena Fuseki does not incorporate any feature of this kind.

- As an alternative to the previous point, also the Database alone could be set up as a Web Service, to represent a centralized data storage endpoint accessible to many users (i.e: employees of a same organization). This choice would be useful for retrieving data from distributed sources, but still would lead to the same problems previously explained, even more emphasised because of the multiple scraping processes accessing to the same site and many multiple requests to the same DB.

- An improvement of the GUI could be relevant in terms of the needs of the expert of the

financial domains, giving the finNSEMA system more power, authority and semantics.

In closing, a check from real and renowned experts of financials would surely bring finNSEMA to a practical fulfilment and enhancements we can not be able to figure out would presumably come out. But the contribution of finNSEMA lays the first foundation stone in the semantic analysis of financial news and gives life to a continuation of this epitome, up to the realization of a professional and fully deployed "in production" semantic financial analysis systems.

Bibliography

[1] Software Freedom Conservancy. Selenium. https://selenium.dev.

[2] EDMCouncil. Fibo. https://spec.edmcouncil.org/fibo/.

[3] Francesco Antoniazzi Alfredo D'Elia Cristiano Aguzzi Tullio Salmon Cinotti Fabio Viola, Luca Roffia. Interactive 3d exploration of rdf graphs through semantic planes. volume 10(8). MDPI, 2018.

[4] Apache Foundation. Apache jena-fuseki. https://jena.apache.org/index.html.

[5] Morton Glantz and Robert L Kissell. *Multi-asset risk modeling: techniques for a global economy in an electronic and algorithmic trading era*. Academic Press, 2013.

[6] William L Hamilton, Kevin Clark, Jure Leskovec, and Dan Jurafsky. Inducing domain-specific sentiment lexicons from unlabeled corpora. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing. Conference on Empirical Methods in Natural Language Processing*, volume 2016, page 595. NIH Public Access, 2016.

[7] keras team. keras. https://keras.io.

[8] KPMG. Pulse of fintech h1'19 – global trends. https://home.kpmg/xx/en/home/campaigns/2019/07/pulse-of-fintech-h1-19-global-trends.html.

[9] Riverbank Computing Limited. Pyqt. https://wiki.python.org/moin/PyQt.

[10] nltk. Nltk. https://www.nltk.org/.

[11] Paralleldots. Paralleldots. https://www.paralleldots.com/.

[12] Nassim Nicholas Taleb. *The Black Swan: The Impact of the Highly Improbable*. Random House, 2007.

[13] Stanford University. Protégé. https://protege.stanford.edu/.

[14] Wikipedia. Algorithmic trading. https://en.wikipedia.org/wiki/Algorithmic_trading.

[15] Wikipedia. Thomson reuters business classification. https://en.wikipedia.org/wiki/Thomson_Reuters_Business_Classification.

List of Figures

2.1 finNSEMA functional architecture. 5
2.2 finNSEMA class diagram 6
2.3 finNSEMA ontology graph representation. 8
2.4 finviz.com news portal. 10
2.5 Web scraper activity. 10
2.6 Triples Builder activity. 12
2.7 finNSEMA storage organization. 12
2.8 Knowledge base extraction. 14
2.9 The upper plane contains filtered data according to "concepts related to positive news". 15
2.10 The upper plane contains filtered data by Companies 15
2.11 The upper plane contains filtered data by "negative concepts" 16
2.12 Another perspective view 17
2.13 finNSEMA GUI 18
2.14 Query generation panel GUI. 19

List of Tables

2.1 finNSEMA technologies. 4