Contents

1. Introduction 2
2. Objectives 2
3. Requirements 2
   3.1. Functional requirements 3
   3.2. Nonfunctional requirements 4
4. System structure 4
   4.1. LOD system structure 4
      4.1.1. Profiling system 4
      4.1.2. Dual use system 5
5. Neural networks 6
   5.1. Convolutional neural networks 7
   5.2. Feed forward neural networks 8
6. Difficulties 8
7. Quora 9
8. Techniques 9
9. User manual and Interface 10
10. User Manual for backend 15
   10.1. Neural network 15
   10.2. Database – MySQL 16
   10.3. The virtual machine: 17
   10.4. The project 17
11. Conclusion 18
12. References 19

1. Introduction

Question Answering systems are getting smarter day by day and most of them are designed to answer questions that are posed in natural language, in some cases questions might be malicious or dangerous in the sense that they might imply a threat or an offensive act; for example killing someone or bombing a place etc.

Our project come into place to address these issues, it makes use of datasets such as DBpedia and then use their content to evaluate the question and identify the level of dangers according to some specific categories, it also make use of neural networks for text classification which helps overcome some problems related to the semantic of the sentence, in addition, Web scraping techniques were used to validate the functionality of the system against data sets such as Quora.

2. Objectives

Developing a plugin system that is able to receive questions from QA system both in natural language and as an RDF and then do processing to decide whether the question is malicious or not and whether it should be answered or not. The system will also provide some statistics between the different results from different data sets and will also provide reports to indicate questions that it considers dangerous. The system operates on different datasets such as DBpedia and Quora.

3. Requirements

A *software requirements specification* (SRS) is a description of a software system to be developed. It lays out functional and non-functional requirements and may include a set of use cases that describe user interactions that the software must provide [1].

1. Functional Requirements

   A description of the facility or feature required. Functional requirements deal with what the system should do or provide for users. They include a description of the required functions, outlines of associated reports or online queries, and details of data to be held in the system.

2. Non-Functional Requirements

   A description, or possible target values, of the associated non-functional requirements. Non-functional requirements detail constraints, targets or control mechanisms for the new system [2].

   Our system includes both functional and nonfunctional requirements.

3.1. Functional requirements

4. Profiling

   Extracting from the knowledge base the words which belong to malicious topics and create database for profiles

5. Dual use system

   - Identify the malicious words in the query
   - Tokenization of the question.

     Breaking the question into words and eliminating the trivial words.
   - Query the tokens.

     Querying the profiles for each word and calculating the malice degree of the question.
   - Define and Test Dual-Use Metrics.
   - Firing the query on DBpedia.
     - Comparing the resources from DBpedia with resources in malicious profiles.

- Calculate the question fitness ratio.

- Block answer of malicious queries

  - Comparing the fitness ratio to a predefined threshold.

  - Take action according to the fitness ratio.

- Create quality Report.

  Generate report with statistics about the question.

- Create a log for malicious queries.

  Logging the malicious queries.

3.2. Nonfunctional requirements

- An efficient system ensuring high performance for large datasets.
- Due to using technologies as RDF and HashMaps.
- A well documentation of navigational files.
- Setup configurations, readme, other third party tools or libraries used.

4. System structure

4.1. LOD system structure

Our system consists of two units: the profiling system and the dual use system. We will explain both of their structure in this section.

4.1.1. Profiling system

The profiling system is an offline process where we create the database that contains malicious words and resources categorized under different classes. The resources are taken from multiple sources such as DBpedia and WikiData. The structure of the profiling system consists of:

1. Defining malicious categories

   We defined 8 categories chosen from the most malicious topics found on the internet. Defining these categories was a manual process. The categories that we choose were: military, nuclear, terrorism, weapons, technology, security, harm and suicide.

2. Finding synonyms and ranking them

   We defined the malicious words for each category by two different methods. In the first method, we defined the words manually and stored them under their categories, in the second method we used *thesaurus* API to find all the related synonyms for each word in the categories.

   We ranked the malicious words using the *uclassify* API. This API rank the words if it is positive or negative.

3. Querying Synonyms

   Querying the synonyms from each category in Dbpedia and Wikidata to get all the resources related to each word.

4. Building database and Filing categories

   Creating the profiling system database which consists of the categories table, the words table, the DBpedia table and the Wikidata table.

   After that, we stored all the resulted resources in the database each in its corresponding table.

4.1.2. Dual use system

1. Tokenize query

   Tokenizing the natural language question. And removing the trivial words.

2. Querying tokens

Querying our profiles database for each word from the tokenized words and calculate the overall ranking of the question. This ranking will be compared to predefined threshold and marked as malicious or not. If it is malicious the system will forward it to dual use system.

3. Firing against LOD sources

   The RDF query is fired against DBpedia and Wikidata to get the related resources.

4. Ranking Query

   In this step, we calculate the ratio between the number of occurrences of The resources from the previous step in the profiles database and their overall number.

5. Block response

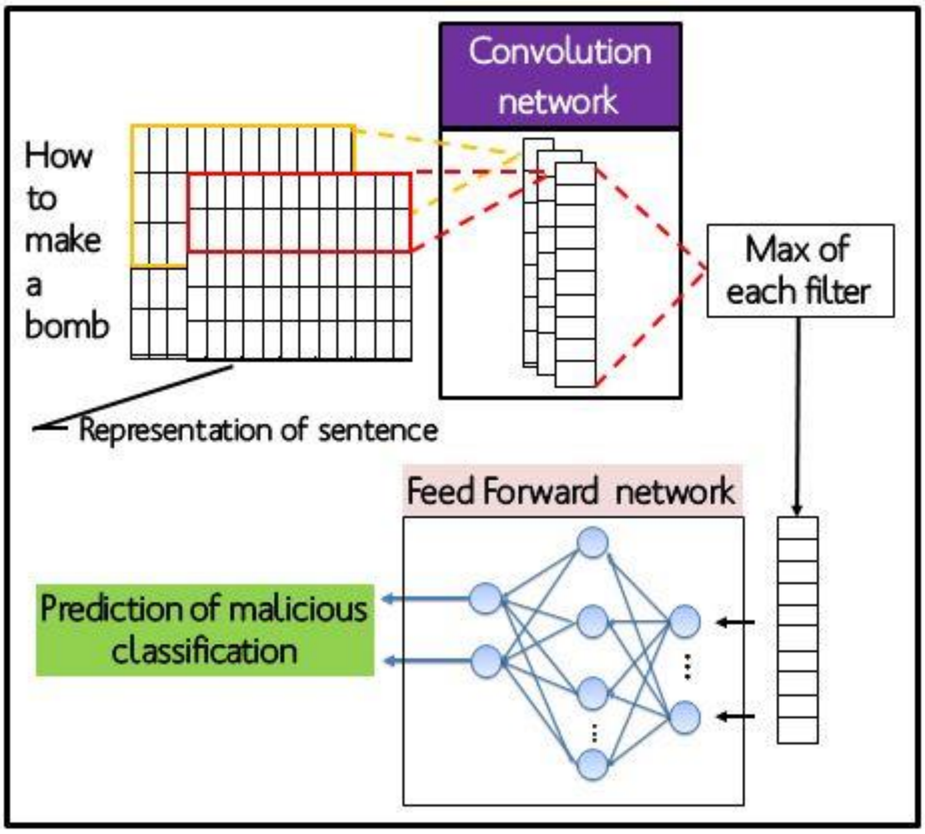
   If the ratio prom the previous step exceeds a predefined threshold (0.5) the system prevents the Question answering system from giving an answer.

6. Generate quality report

   Generate a report with statistics about the question.

5. Neural networks

As discussed in the Difficulties section, we faced problems related to the semantics of the question, where 2 questions with varying danger levels would have a very close ranking and therefore they would be treated in the same manner, as a result we made use of supervised neural networks which can learn and improve according to the data fed in in order to classify the different levels of danger between questions in a more accurate way. We used an approach that is similar to the one published in the paper[3] where they used Convolutional Neural Networks for Sentence Classification. In this section, we will give a brief explanation of this approach. Fig1. shows the complete neural networks model.



5.1. Convolutional neural networks

The model architecture, shown in Fig.1, contains a CNN architecture. The CNN input is the k-dimensional word vector corresponding to the i^{th} word in the sentence. A sentence of length N is presented as a concatenation of words. A convolution operation involves a filter, which is applied to a window of words to produce a new feature. This filter is applied to each possible window of words in the sentence to produce a feature map. We then apply a max-overtime pooling operation over the feature map and take the maximum value as the feature corresponding to this particular filter. The idea is to capture the most important feature—one with the highest value—for each feature map. This pooling scheme naturally deals with variable sentence lengths. This is a description of the process by which one feature is extracted from one filter. The model uses multiple filters (with varying window sizes) to obtain multiple features.

5.2. Feed forward neural networks

the features from the previous step are passed to a fully connected softmax layer whose output is the probability distribution over labels

6. Difficulties

- Ranking of the words: We first tried to use *SentiWordNet* API to rank the words in the categories but the results were inaccurate so we resorted to using *uclassify* API which gave much better results but not as satisfactory as we wished. As we wanted to rank according to how malicious it is but we didn’t find any tool to help us.
- Semantic of the sentence in the threshold-based approach. Basically, when using this approach, the question is tokenized and then fired against the database and against the profiles we have in there, therefore sentences such as: How to kill a person VS. How to kill an insect, usually result in the same danger level although the first one is more dangerous. That’s why we made use of neural networks which are able to learn in a supervised way and then improve and eventually realize the semantic difference between these two.
- The content and information on DBpedia and Wikidata is not structured well and also these two datasets don’t really contain much controversial topics and categories, therefore we had to implement tests on Quora in order to validate the task of our

system, as it was hard to do that on the previously mentioned datasets because of their limited content regarding malicious categories.

- Up to this date, the QA-system and the system which is supposed to provide our system with both natural language questions and RDF questions are not present.

7. Quora

As we had some problems with the structure and data on Wikidata and DBpedia, we used Quora as a dataset to test our system on.

Our system provide a tool that implements some ‘Web scraping’, basically it uses the question asked in its natural language form and then fire it against google and fetch the answers available from Quora, after that it gets the answers posted for that particular question and apply the techniques such as ranking and tokenization which make use of profiling to rank the danger level of the answers and therefore the question itself, so in brief, it applies the same techniques used on the fed in questions, but this time on Quora.

8. Techniques

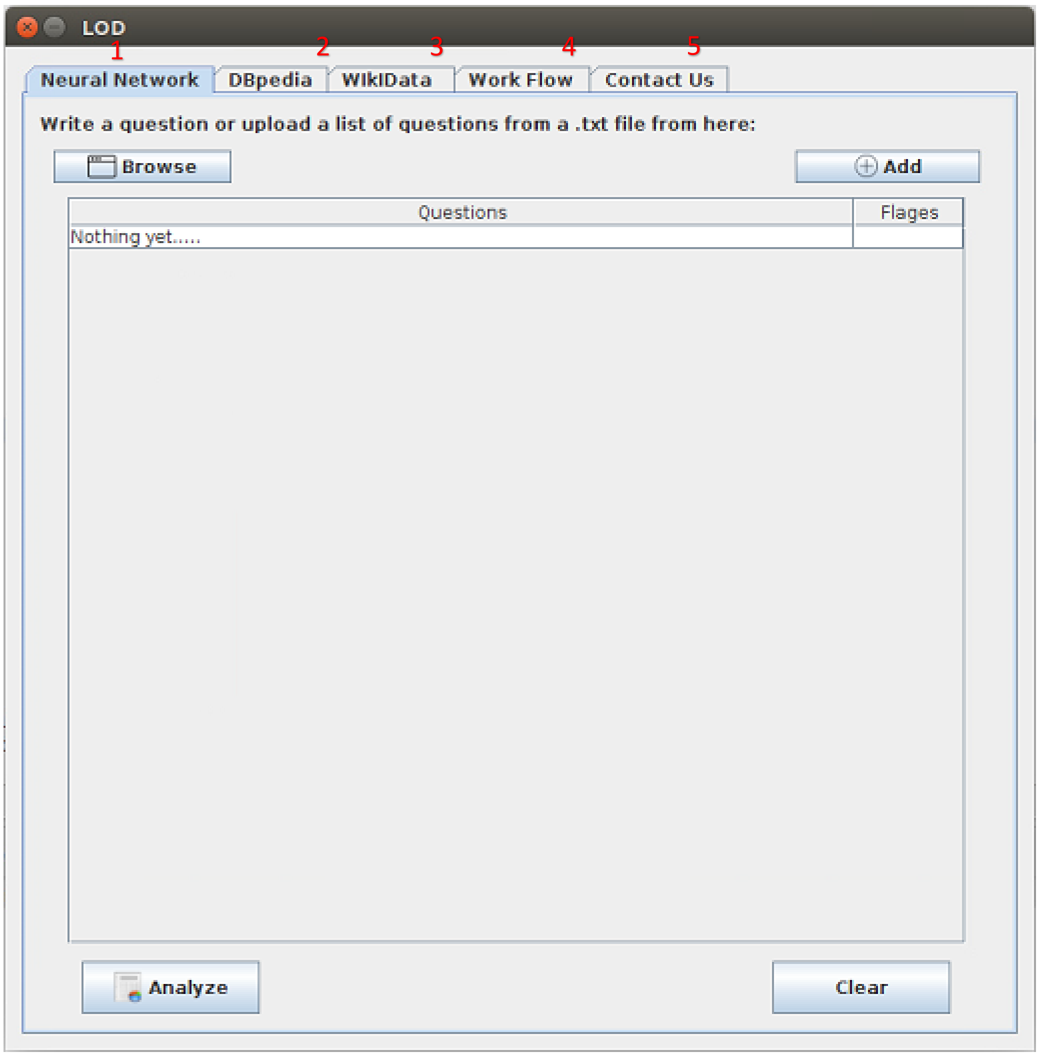
- **JAVA:** The main programming language used for coding the project.
- **MySQL:** To build and connect to the profiles database.
- **Jena :** Open source Semantic Web framework for Java , We used it to extract data from the LOD.
- **Snowball:** Small string processing language which provides us with the root of the potentially malicious words’ root to add to our database.

- **TensorFlow :** An open source software library .we used it to build the deep neural network.
- **Python :** The programming language we used to write the code for building and training the deep neural network using the TensorFlow library.
- **Jsoup:** A java library that is used to parse HTML document. As we needed to tokenize the web pages to search for malicious words and rank it as malicious or not.
- **JFreeChart:** An open-source framework for the programming language Java, which allows the creation of charts based on the comments in the code. This was used for the documentation.

9. User manual and Interface

The program will contain four main tabs:

- Neural Network 1
- DBpedia 2
- Wikidata 3
- Workflow 4
- And an addition a contact us tab 5

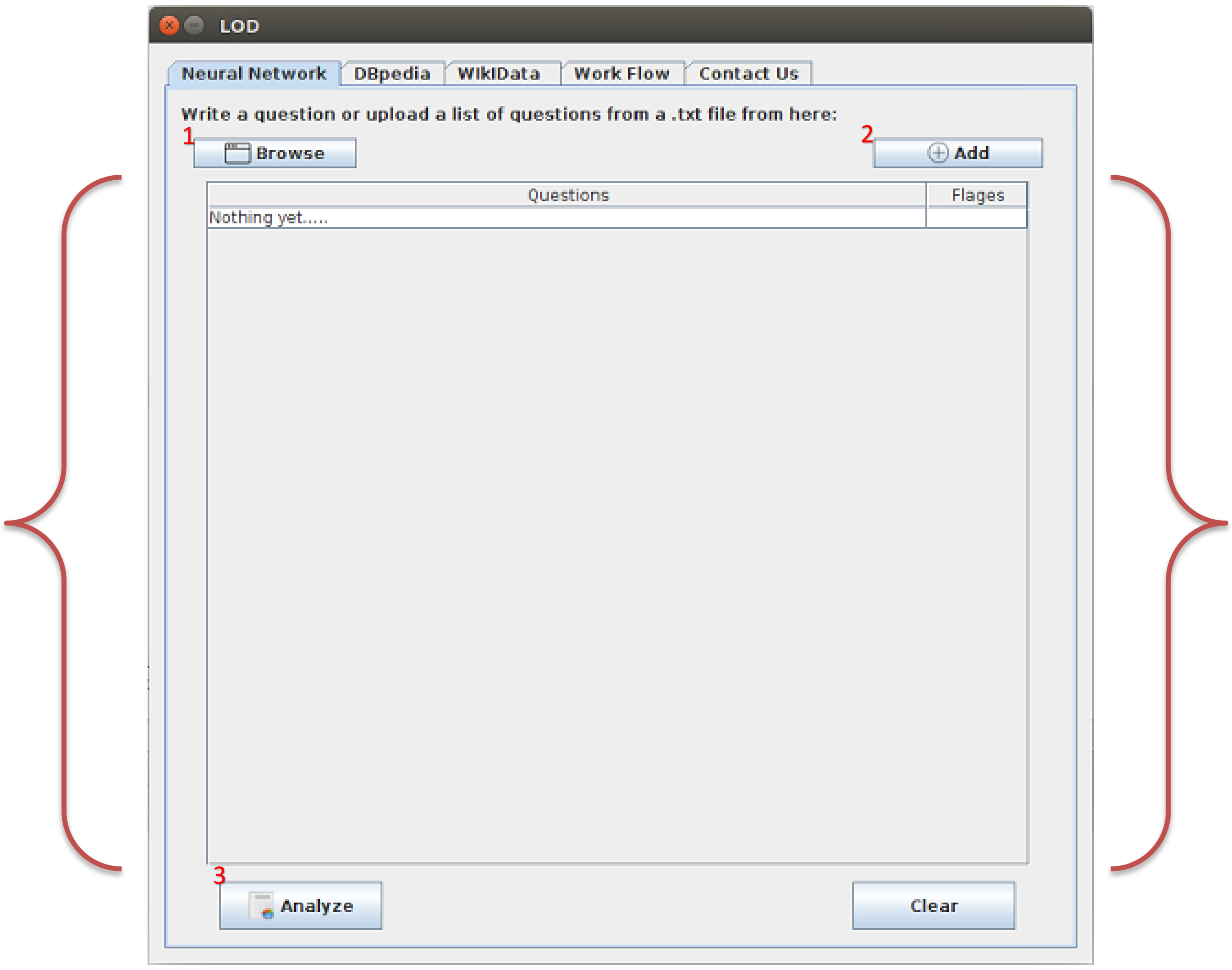


Write a question or upload a list of questions from a .txt file from here:

Nothing yet.....

Neural Network tap:

- Browse button: to choose a bunch of question from .txt file 1
- Add button: to add single question 2
- Analyze: to run the neural network analysis and get the results 3
- Clear: to delete the questions from the main table 4
- Main table: this table will show the questions their classifier 5



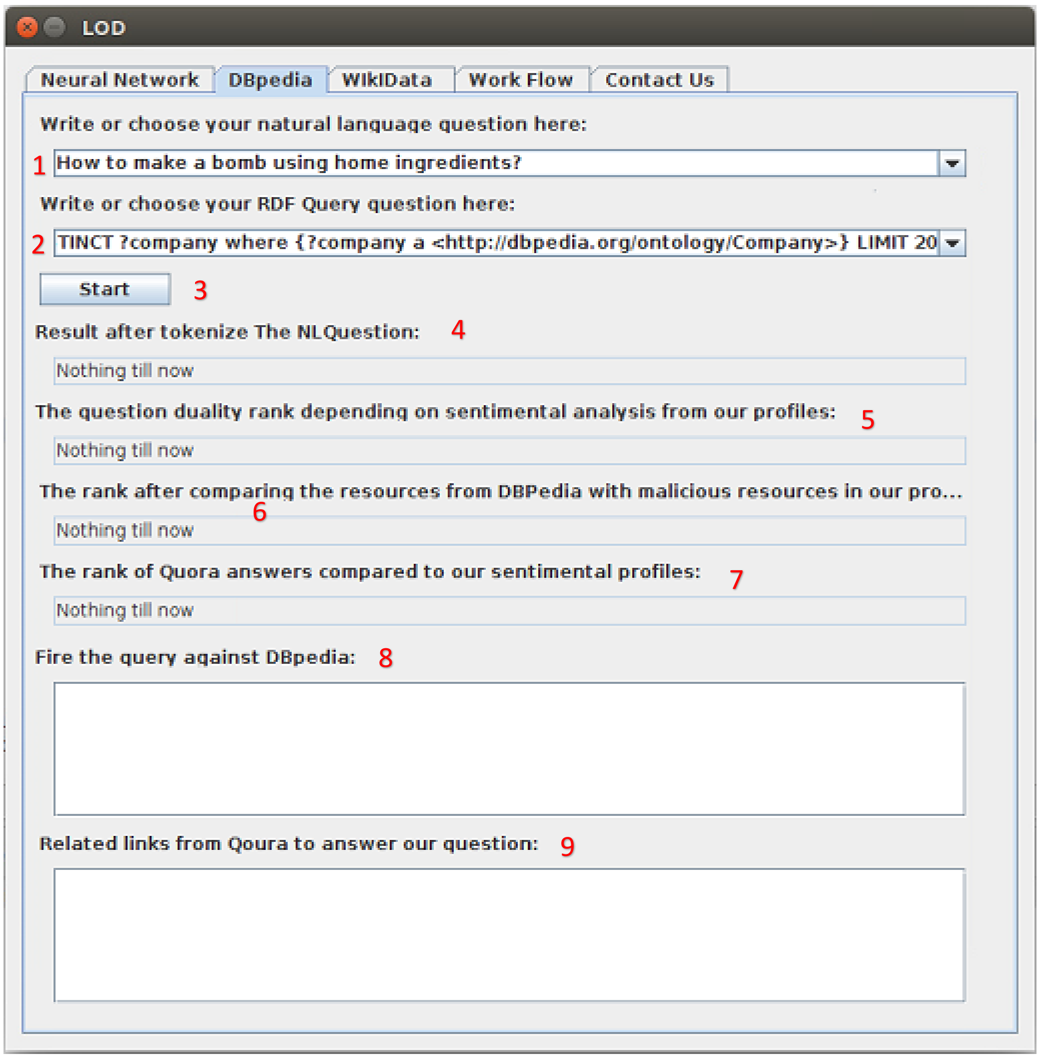
Write a question or upload a list of questions from a .txt file from here:

Nothing yet.....

DBpedia tap:

- Natural language question: to write or choose a question in natural language form 1
- SPARQL query: to write or choose a question in RDF (SPARQL) 2
- Start button: to start analyze the questions 3
- Tokens: here the question after tokenizing 4
- First rank: the rank produced depending on sentimental analysis 5
- Second rank: the rank produced depending on DBpedia resources 6
- Third rank: the rank produced depending on sentimental analysis of Quora results 7

- DBpedia resources: the resources from Dbpedia 8
- Quora resources: the resources from Quora 9



Write or choose your natural language question here:

How to make a bomb using home ingredients?

Write or choose your RDF Query question here:

TINCT ?company where {?company a <http://dbpedia.org/ontology/Company>} LIMIT 20

Result after tokenize The NLQuestion:

Nothing till now

The question duality rank depending on sentimental analysis from our profiles:

Nothing till now

The rank after comparing the resources from DBPedia with malicious resources in our pro...

Nothing till now

The rank of Quora answers compared to our sentimental profiles:

Nothing till now

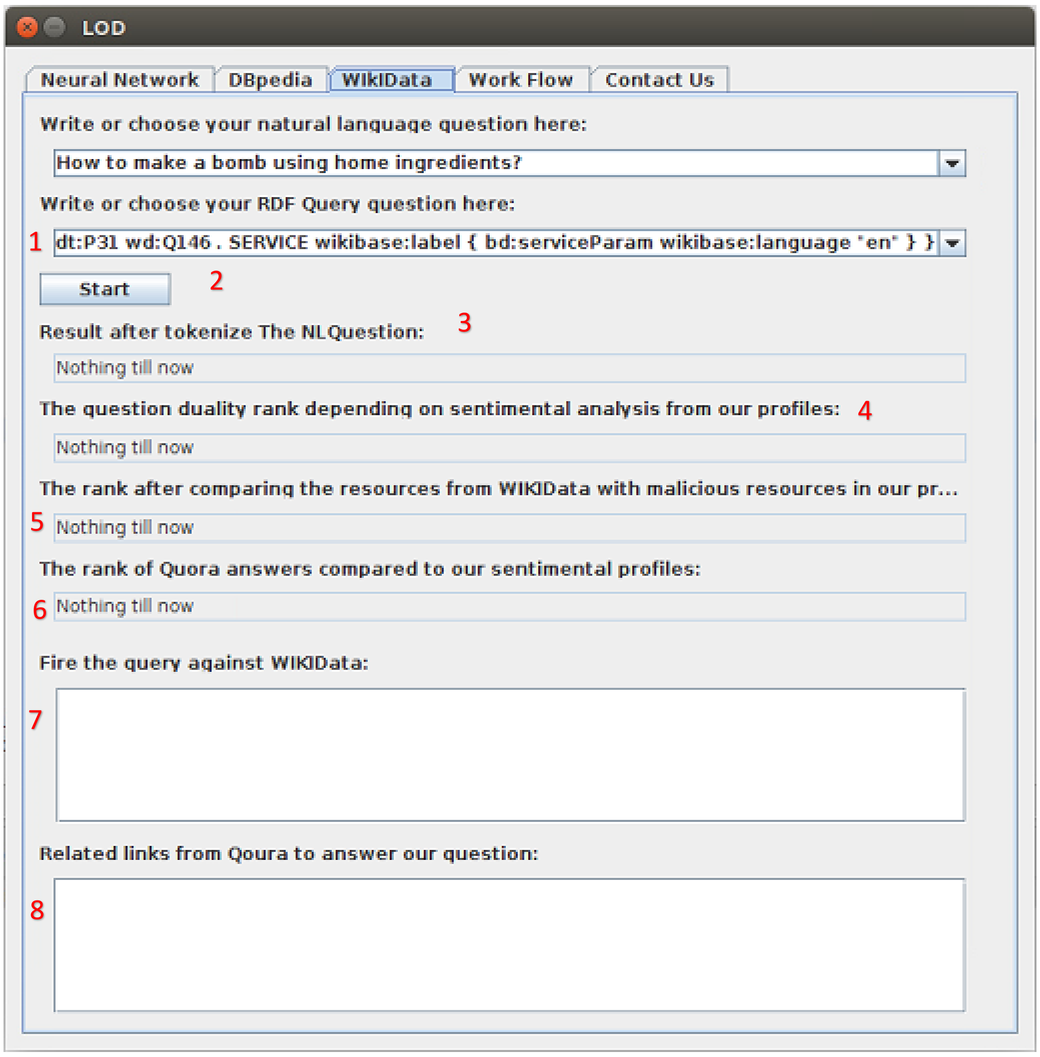
Fire the query against DBpedia:

Related links from Qoura to answer our question:

Wikidata tap:

- Natural language question: to write or choose a question in natural language form
- SPARQL query: to write or choose a question in RDF (SPARQL) 1
- Start button: to start analyze the questions 2
- Tokens: here the question after tokenizing 3
- First rank: the rank produced depending on sentimental analysis 4
- Second rank: the rank produced depending on Wikidata resources 5

- Third rank: the rank produced depending on sentimental analysis of Quora results 6
- DBpedia resources: the resources from Wikidata 7
- Quora resources: the resources from Quora 8



Write or choose your natural language question here:

How to make a bomb using home ingredients?

Write or choose your RDF Query question here:

dt:P31 wd:Q146 . SERVICE wikibase:label { bd:serviceParam wikibase:language "en" } }

Result after tokenize The NLQuestion:

Nothing till now

The question duality rank depending on sentimental analysis from our profiles:

Nothing till now

The rank after comparing the resources from WIKIData with malicious resources in our pr...

Nothing till now

The rank of Quora answers compared to our sentimental profiles:

Nothing till now

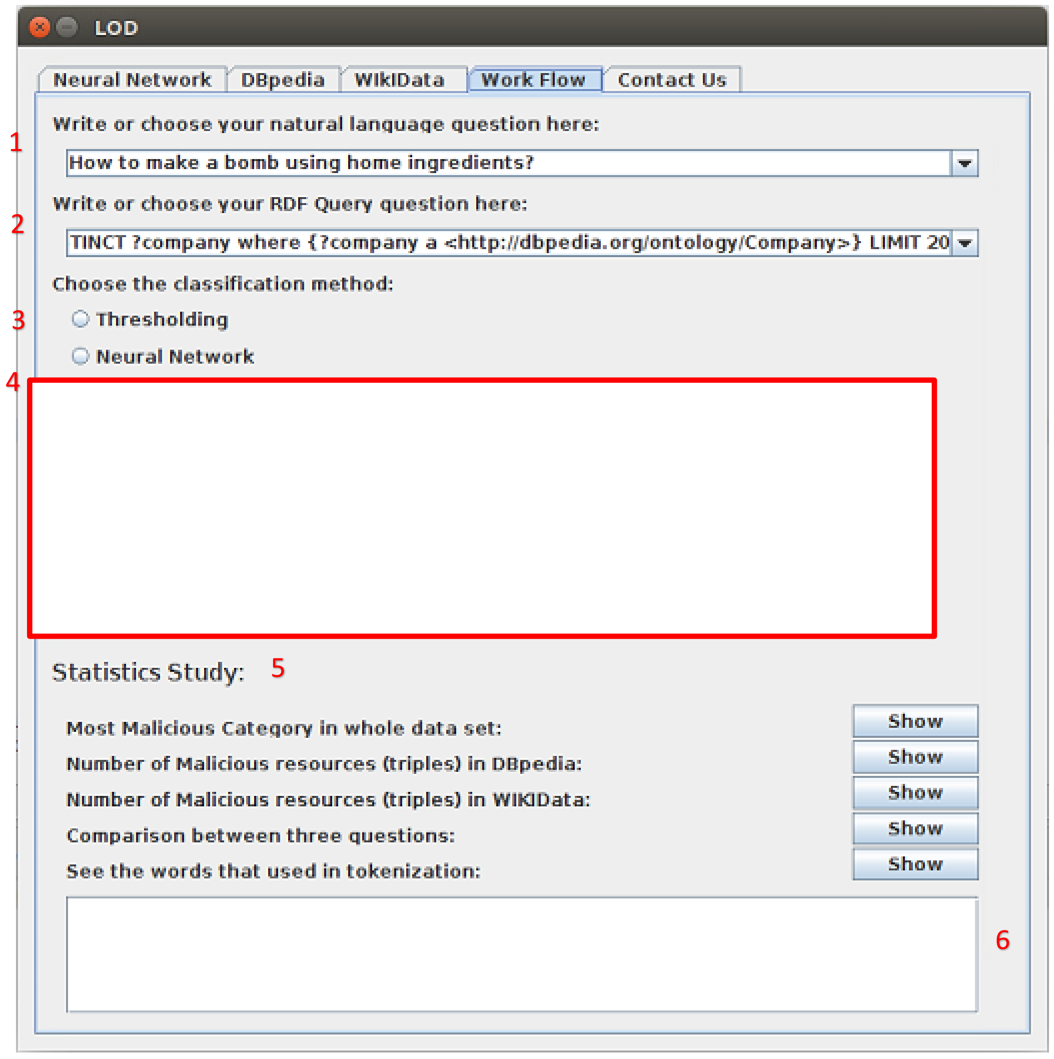
Fire the query against WIKIData:

Related links from Qoura to answer our question:

Workflow tab:

- Natural language question: to write or choose a question in natural language form 1
- SPARQL query: to write or choose a question in RDF (SPARQL) 2
- Radio button: to choose the method of classification 3

- Blank area to show the user the results and ranks from Dbpedia and Wikidata 4
- A bunch of statistical studies depending on the constructed database 5
- Show the words used in tokenization with the able of adding or deleting 6



Write or choose your natural language question here:

How to make a bomb using home ingredients?

Write or choose your RDF Query question here:

TINCT ?company where {?company a <http://dbpedia.org/ontology/Company>} LIMIT 20

Choose the classification method:

Thresholding

Neural Network

Statistics Study:

Most Malicious Category in whole data set:

Number of Malicious resources (triples) in DBpedia:

Number of Malicious resources (triples) in WIKIData:

Comparison between three questions:

See the words that used in tokenization:

10. User Manual for backend

10.1. Neural network

The neural network is located at:

~/home/deep-learning/workspace/CNN_Classification when you wish to train the network, the file train.py will do the job so basically you need to run the command:

```
python train.py
```

Which will create a new trained run in the runs file in the same directory.

```
~/home/deep-learning/workspace/CNN_Classification/data/rt-polaritydata
```

In this directory, you can find two files one related to positive questions and the other to negative questions. Positive_Questions , Negative_Questions. You can append or alter these files to retrain the network. Note that then in the LOD project you need to alter the neural networks profile to be loaded: now navigate to the *LOD project and the LOD.java* class and alter the following command according to the run you already created, alter the underlined run-number:

```
Process p = Runtime.getRuntime().exec("python /home/deep-
learning/workspace/CNN_Classification/test.py /home/deep-
learning/workspace/CNN_Classification/runs/1476970477/vocab
/home/deep-
learning/workspace/CNN_Classification/runs/1476970477/checkpoi
nts");
```

10.2. Database – MySQL

You can check MySQL-Workbench and navigate to the database *Gdus* that contains the following tables:

- The Category has the malicious categories.
- The Word has the words related to the malicious categories in addition to a rank calculated using sentiment analysis.

- DBpedia data has the word id and its corresponding resource from DBpedia.
- Wikidata has the word id and its corresponding resource from Wikidata.

10.3. The virtual machine:

When you access the virtual machine and open eclipse you will find 3 projects:

- LOD: which is the main project and has most of the functionality
- CNN_Clasification: which contains everything regarding the neural networks
- Gdus: which contains the classes responsible for the profiling

10.4. The project

Our project was created in the first place as a *plug-in* to a *QA system* in order to assess the questions, so when you navigate to the project *LOD*, you will find that the program has a GUI, but if you wish to use it as a plug-in the whole functionality can be found in the class: *LOD.java*.

Basically, you will feed the program two parameters:

1. Natural language question
2. The corresponding SPARQL-query

and then you will receive the rank our program computed for these questions. You can find a method in *LOD.java* class, that is called: *LODSys* in which you should pass 3 parameters:

1. The first parameter: 1 if you wish to use neural networks, 2 if you want to use thresholding based ranking.
2. The Second parameter: The question in natural language.
3. The Third parameters: The SPARQL query.

In case the question was not malicious, your result will be null, otherwise, the result will be a list of two results, the first one is the rank computed after getting the Wikidata results and comparing them to the profiles and the second one is the rank computed after getting the DBpedia results. So, checking the size is advisable because if the size of the returned list is 0 so your question is not marked as malicious according to the technique used.

11. Conclusion

Linked open data resources have a great dual use potential if structured correctly. As it is now the data are available but not structured to form a complete knowledge that could be used in malicious ways. But in all cases, the internet is full of information that could be used in a malicious way and a monitoring and restricting mechanism as the one used in our project is a huge need in order to protect such public data.

12. References

[1] Bourque, P.; Fairley, R.E. (2014). "Guide to the Software Engineering Body of Knowledge (SWEBOK)". IEEE Computer Society. Retrieved 17 July 2014.

[2]http://www.sqa.org.uk/e-learning/SDM03CD/page_02.htm#FuncNonFuncReq

[3] Kim, Y. (2014). Convolutional Neural Networks for Sentence Classification. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP 2014), 1746–1751.