Third International Conference
Modelling and Development of Intelligent Systems
October 10 - 12, 2013
"Lucian Blaga" University
Sibiu - Romania

Detecting influenza epidemics based on real-time semantic analysis of Twitter data

Radu Balaj, Adrian Groza

Abstract

The paper presents a method for detecting influenza rates in a geographical region, using the knowledge extracted from Twitter messages. The novelty consists of using fuzzy description logic for linking natural language in Twitter streams with formal reasoning in description logic. In order to analyze the data in a medical context, the content of a Twitter post is transformed into the RDF stream format, and queried using the C-SPARQL query language. The conducted research lies in the larger context of semantic stream reasoning.

1 Introduction

The large-scale popularity of social platforms such as Facebook or Twitter makes them suitable for applications which analyze heterogeneous data from multiple users in real-time. One of the areas which can benefit from the nature of the data generated by these platforms is the public health domain.

1.1 Motivation

Public health is concerned with the prevention of diseases that might significantly impact certain communities of individuals. Traditional methods of disease surveillance, such as analyzing data provided by medical laboratories, are time-consuming and can result in a late detection of a disease epidemic by as much as two weeks [1]. Different alternative methods have been proposed in the literature, which take advantage of the real-time, huge amount of data made available on the Internet.

A question that this research addresses is *"Are the rates of influenza indicative of an epidemics outbreak in region X¿*'. The choice of influenza is due to its large distribution and to its very familiar symptoms. Thus, it is much more likely that Twitter users will generate relevant data for the detection of influenza, more than for any other disease.

1.2 Technical challenge

An application which aims to monitor a great amount of real-time data has its own challenges. One of the recent areas of research which deals with such systems is stream reasoning [2]. Its main objective is the generation of new tools and methods which integrate data streams, the Semantic Web and reasoning systems. Therefore, these new approaches could tackle new and complex problems [8], which can not be solved using just one or two of those areas.

Stream reasoning is most suitable for applications which use data that is:

- *rapidly changing*: most reasoning system use a static knowledge base, which needs an update process in order to integrate new data.
- *of great dimensions*: the data is too big and/or too expensive to store.
- *heterogeneous*: the data itself can be inconsistent, as in the case of data provided by sensors (transmission errors) or extracted from multiple ontologies (difficulties integrating concepts).

Thus, considering the nature of the problem at hand and the choice of Twitter as the data source, the analysis of tweets for influenza detection is suitable to the domain of stream reasoning. The information provided by Twitter is rapidly changing (there are thousands of tweets generated every minute), of great dimensions (it is impractical to store messages) and it is heterogeneous (because the content of a message is susceptible to interpretation errors).

The remainder of this paper is structured as follows. The following section briefly describes the technical solution, before detailing the technical and theoretical background. Section 3 presents important aspects from the implementation of the system. Section 4 describes relevant experimental results, along with a discussion regarding these. The last two sections discuss similar approaches proposed in the literature and present the final conclusions of this research.

2 Technical and theoretical background

This section describes the relevant technical and theoretical aspects of the methods and tools used in the actual implementation of the solution. First of all, it is suitable to offer a brief overview of how these were used so that the reader can put them in the right context.

2.1 Short overview

The posts generated on the Twitter platform are collected by the application using the Twitter Streaming API. Thus, around 1% of the total number of messages generated at one time is available in the application. In order to process just the relevant messages, the content of a tweet is matched against a collection of relevant words (that either describe a symptom or mention flu/influenza directly). The Stanford Dependencies parser is used in order to provide the grammatical relationships of the words in a message, which helps at further discarding irrelevant tweets (by eliminating negative tweets). The result after the parsing phase is a set of symptoms and the corresponding attributes, if any. The symptoms which are checked come from a knowledge base in the form of an ontology. This ontology, called Symptom Ontology, is further refined in order to entail fuzzy notions of symptoms manifestation, using the fuzzyDL (fuzzy Description Logic) syntax. The protocol for detecting influenza in a Twitter message is a set of rules which are interpreted by the fuzzyDL reasoning engine. After applying the protocol, a message is transformed in the RDF stream format. A C-SPARQL query over the generated RDF stream gives the number of influenza cases detected in a certain period of time, for a certain geographical region. Figure 1 outlines the generic architecture, with its main components.

2.2 Technical background

The tools and methods mentioned in the overview are briefly described in this section.

Twitter Streaming API. This API[1] allows an application to connect to a Twitter streaming endpoint, which feeds the messages to the application, without the need to constantly make explicit requests. A

[1] https://dev.twitter.com/docs/streaming-apis

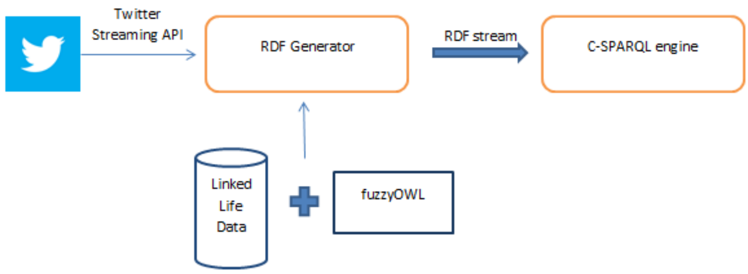

Figure 1: The architecture of the proposed solution.

random feed of messages is available through the $GETstatuses/sample$ endpoint. From the total number of messages, a percentage of 1% is available at any time.

Stanford Dependencies parser. Parsing a message in natural language is a difficult task. The SD parser [3] was built in order to provide a description of the grammatical relationships in a sentence, in a way that is understood by non-specialists. The main design principles of the parser is the representation of grammatical relationships through pairs of words and the choice of using traditional notations for those relationships. A parsed sentence has a hierarchical tree form, with the most generic relationship as the root of the tree.

Linked Life Data ontologies. Ontologies are a formal method of representing the knowledge from a domain. Since the functionality of the proposed solution is part of the medical domain, it is suitable to make use of existing ontologies. The Linked Life Data repository [4] is a platform for the integration of medical and biological data. It is comprised of more than 20 ontologies, one of which is of great use for this solution. Symptom Ontology (SO) is a hierarchical ontology which entails symptoms according to the body region in which they arise. These symptoms can be linked with concepts from other knowledge bases such as the Disease Ontology. We make use of SO for describing the influenza symptoms.

FuzzyDL. The fuzzyDL reasoner[2] is a Description Logic reasoner which supports fuzzy logic reasoning, by extending the classic description logic SHIF to the fuzzy case. In order to create or modify ontologies for fuzzyDL, a plug-in for the ontology editor Protege can be used, called fuzzyOWL2[3]. This plugin was used to add fuzzy notions to the already existing Symptom Ontology.

C-SPARQL. In order to interpret the knowledge from an ontology, the SPARQL query language is used. However, SPARQL can only query on static knowledge. Hence, the C-SPARQL [5] extension was designed in order to query over dynamic data in the RDF stream format. Thus, the language is extended to include physical (number of RDF triples) and logical (time-related) windows over which the query is applied.

Operation	Lukasiewivz Logic	Gödel Logic
intersection	$\alpha \otimes_L \beta = max\{\alpha + \beta - 1, 0\}$	$min\{\alpha, \beta\}$
union	$\alpha \oplus_L \beta = min\{\alpha + \beta, 1\}$	$max\{\alpha, \beta\}$
negation	$\alpha \ominus_L \beta = 1 - \alpha$	$\alpha \ominus_L \beta = 1,\ if\ a = 0,\ 0,\ otherwise - \alpha$
implication	$\alpha \Rightarrow_L \beta = min\{1, 1 - \alpha + \beta\}$	$1, if \alpha \leqslant \beta, \beta, otherwise$

Figure 2: Operators in Fuzzy Logics.

2.3 Theoretical aspects

Fuzzy Description Logic (FDL) has been proposed as an extension to classical description logic with the aim to deal with fuzzy and imprecise concepts, and it is based on the *SHIF(D)* version of the description logic. Some observations regarding the appliance of the fuzzy operators (figure 2) to argumentation follow:

The interpretation of Gödel operators maps the *weakest link principle* in argumentation. According to this principle, an argument supported by a conjunction of antecedents (α,β,...) is as good as the weakest premise ($min\{\alpha, \beta, ...\}$). Similarly, when several reasons are used to support a consequent, the strongest justification is chosen ($max\{\alpha, \beta, ...\}$). From the nature of the argumentative process itself, the subject of the debate cannot be easily categorised as true or false. The degree of truth for an issue and its negation ($1 - \alpha$) are continuously changed during the lifetime of the dispute. Thus, the different levels of trueness (and falseness) from fuzzy logic can be exploited to model argumentation.

The interpretation of Lukasiewics operators fits better to the concept of *accrual of arguments*. In some cases, independent reasons supporting the same consequent provide stronger arguments in favor of that conclusion ($min\{\alpha + \beta, 1\}$). Similarly, several reasons against a statement act as a form of collaborative defeat [7].

In the following paragraphs the differences introduced by fuzzy reasoning on top of classical description logic are presented. The complete formalisation of the fuzzy description logic can be found in [6].

A fuzzy knowledge base $K = \langle A, T, R \rangle$, consists of a fuzzy ABox *A*, a fuzzy TBox *T* and a fuzzy RBox *R* [6]. A fuzzy ABox A consists of a finite set of assertion axioms for fuzzy concepts $\langle x : C, \alpha \rangle$, and fuzzy roles $\langle (x, y) : R, \alpha \rangle$, where $\alpha \in [0, 1]$, C is a concept, and R a role. For instance, $\langle david : SmallPerson, 0.8 \rangle$ states that $david$ is a SmallPerson with degree at least 0.8, whilst $\langle (david, goliat) : attack, 0.7 \rangle$ says that *david* has attacked *goliat* with degree at least 0.7. If α is omitted, the maximum degree of 1 is assumed.

A fuzzy TBox T is a finite set of inclusion axioms $\langle C \sqsubseteq_S D, \alpha \rangle$, where $\alpha \in [0, 1]$, C, D are concepts, and S specifies the implication function (Lukasiewicz, Gödel) to be used. The axioms state that the subsumption degree between C and D is at least α.

A fuzzy RBox is a finite set of role axioms of the form: $(fun\ R)$, stating that the role R is functional; $(trans\ R)$, stating the role R is transitive, $R_1 \sqsubseteq R_2$, meaning the role R_1 is subsumed by the role R_2; and $(inv\ R_1\ R_2)$, stating the role R_1 is the inverse of the role R_2.

The main idea of *semantics* of FDL is that concepts and roles are interpreted as fuzzy subsets of an interpretation's domain [6]. A fuzzy interpretation $I = (\Delta^I, \bullet^I)$ consists of a non empty set Δ^I (the domain) and a fuzzy interpretation function $\bullet^I$. The mapping $\bullet^I$ is extended to roles and complex concepts as specified in figure 3.

[2]http://gaia.isti.cnr.it/straccia/software/fuzzyDL/intro.html
[3]http://gaia.isti.cnr.it/straccia/software/FuzzyOWL/index.html

$\perp^I(x) = 0$	$(\forall R.C)^I(x) = inf_{y \in \Delta^I} R^I(x,y) \Rightarrow C^I(y)$
$\top^I(x) = 1$	$(\exists R.C)^I(x) = sup_{y \in \Delta^I} R^I(x,y) \otimes C^I(y)$
$(\neg C)^I = \ominus C^I(x)$	$(\forall T.d)^I(x) = inf_{y \in \Delta^I} R^I(x,v) \Rightarrow d^I(y)$
$(C \sqcap_S D)^I(x) = C^I(x) \otimes_S D^I(x)$	$(\exists R.C)^I(x) = sup_{y \in \Delta^I} R^I(x,v) \otimes d^I(y)$
$(C \sqcup_S D)^I(x) = C^I(x) \oplus_S D^I(x)$	$(x : C)^I = C^I(x^I)$
$(C \rightarrow_S D)^I(x) = C^I(x) \Rightarrow_S D^I(x)$	$((x,y) : R)^I = R^I(x^I, y^I)$
$(m(C))^I(x) = f_m(C^I(x))$	$(C \sqsubseteq D)^I(x) = inf_{x \in \Delta^I} C^I(x) \Rightarrow_S D^I(x)$

Figure 3: Semantics of fuzzy concepts.

3 Implementation

This section describes in detail the most significant aspects of the implementation process. The focus here is on the tweet parsing algorithm, the design of the fuzzy ontology and the two reasoning steps.

3.1 Modelling fuzzy symptoms

The advantages of fuzzy logic for the representation of symptoms, such as presented in this work, are numerous. First of all, such a representation yields much more accurate results when trying to detect a disease given a number of symptoms. Secondly, the linguistic variables provide an appropriate method of describing a symptom, in a manner similar to how humans verbally express them (e.g. high fever, mild stomach pain). Last but not least, fuzzy logic applied to medical diagnosis has a long history, given by the amount of literature that can be found on the subject.

First of all, three linguistic variables were created: weak, moderate and strong. Each of these is a datatype, mapped on the [0,1] domain. A data property named *isManifested* was created, with the domain being any symptom in the ontology and the range one of the three fuzzy datatypes. Thus, it can be stated that a symptom has three possible degrees of manifestation.

In order to create relevant rules for detecting influenza in a message, we make use of the medical literature through the research presented in [9]. Table 1 presents some of the data based on which the rules from ontology have been created.

Symptom	Percentage(%)
fever	76
cough	69
fever+cough	82
fever+cough+headache	78
fever+cough+runny nose	81

Table 1: Symptoms and their power of predicting influenza infection.

One rule is designed as an equivalence class in the ontology as shown in table 2. There are nine such classes and therefore nine rules.

In order to make use of these rules, a concept of type Weighted Complex Concept(WCC) is created, called InfluenzaWeightedConcept (IWC). A WCC is a class which is composed of two or more classes, with different weights attached to them. The weighting type chosen is weighted maximum, which means that the class with the greatest weight will be chosen. This matches the Gödel interpretation of fuzzy logic, which states that the strongest justification is chosen when there are several reasons to support a consequent.

Class Name	Rule
I1	fever and (isManifested some strong)
I2	fever and ((isManifested some moderate) or (isManifested some weak)))
I3	cough and (isManifested some strong)
I4	cough and ((isManifested some moderate) or (isManifested some weak)))
I5	(chills and (isManifested some moderate)) or (fever and ((isManifested some moderate) or (isManifested some weak)))
I6	fever and cough and headache
I7	fever and cough
I8	(fever or cough) and (pain or sneezing or weakness or 'runny nose' or 'muscle pain')
I9	headache or pain or sneezing or weakness or 'runny nose' or 'muscle pain'

Table 2: Echivalence classes act as rules for detecting influenza.

Practically, if more than one of the nine rules is active (for example, a Twitter message contains both the fever and cough symptoms), the final value in detecting the influenza probability is set by the active rule with the greatest weight.

The definition for the IWC concept is the following:

$$\begin{aligned} \mathit{InfluenzaWeightedConcept} = \quad & 0.68 * I1 + 0.76 * I2 + 0.69 * I3 + 0.62 * I4 + 0.76 * I5 + \\ & 0.78 * I6 + 0.82 * I7 + 0.81 * I8 + 0.1 * I9 + 1 * I10 \end{aligned}$$

Note that the I10 class has been added by necessity. A WCC with the weighting type maximum needs at least one class with weight 1. I10 will not affect the final result, the term is 0 in all scenarios. The weight for each class was set according to the predictive value of the symptoms they entail. Since this is a fuzzy context, an instance a of class C can have a value between (0,1], which signifies the fuzzy degree to which that instance a is a member of class C. Therefore, in practice, the value associated with the instances of the class IWC can give us the degree to which that individual can be said to be a member of the class which represents positive influenza tweets.

For the fuzzyDL reasoner to compute the degree of an instance a of class IWC, the following query is used: $(min - sat?\ \mathit{InfluenzaWeightedConcept}\ [a])$. This query returns the minimum satisfiability degree of the instance a in the class IWC and gives the probability that a Twitter message, abstracted by instance a, entails relevant information regarding influenza.

3.2 Twitter message parsing

Before parsing a tweet using the Stanford Dependencies method, some filtering needs to be applied. First of all, only Twitter messages in English which have the geolocation property set are used. That is because without the geolocation property, it is impossible to detect influenza rates in a geographical region.

Furthermore, only tweets which contain words indicative of influenza, such as flu or symptoms, are processed by SD. The other ones are discarded. The parsing algorithm which extracts the data needed for the reasoning tasks consists of four important steps.

(1) First of all, vocabularies of words are initialised. These are used to match if a Twitter message has symptoms or words referring to influenza in its body, and to match a symptom's attribute to their corresponding fuzzy datatype (weak, moderate or strong). If a word like flu matched, the symptoms are no longer searched in the message. It is considered that the message entails information regarding influenza, and the protocol for detecting influenza is not used. Otherwise, if one or more symptoms matched, they need to be extracted, along with their possible attributes, and passed to the fuzzyDL reasoner.

```
REGISTER QUERY DetectInfluenza AS
SELECT (COUNT(?s) as ?NoOfTweets)
FROM STREAM < http://myexample.org/streamInfluenza > [RANGE 24 HOUR TUMBLING]
WHERE { ?s < http://myexample.org/streamInfluenza/hasInfluenza/ > ?o.
?s2 < http://myexample.org/streamInfluenza/hasLongitude/ > ?long.
?s3 < http://myexample.org/streamInfluenza/hasLatitude/ > ?lat.
FILTER( ?s = ?s2 && ?s = ?s3
&& lat >= 39 && lat <= 42
&& ?long >= 73 && ?long <= 75 )}
```

Figure 4: C-SPARQL query for determining the number of positive influenza tweets.

(2) In the case the matched words are flu or $influenza$, the result from SD parsing is used to make sure that the meaning of the message is positive, rather than negative, such as in the tweet *"As it turns out, I do not have theflu!"*. To detect a negative tweet, the formula used is:

$$if\ neg(verb, negation)\ and\ dobj(verb, flu) \Rightarrow negative$$

If it is found to be negative, the tweet is discarded.

(3) If the body of the message is matched against one or more symptoms, the grammatical relationship $amod$ determines if a symptom has an attribute that can help model the fuzzy degree of manifestation. For example, parsing the message *"I have a terrible fever, no work for me today."* determines the grammatical relationship $amod(terrible, fever)$. Therefore, the symptom fever has associated a fuzzy degree of manifestation.

(4) If an attribute is found, check against the vocabularies in order to establish to which fuzzy datatype the attribute belongs: weak, moderate or strong. In this case, terrible is matched against the vocabulary of words representing the $strong$ fuzzy datatype.

3.3 Stream reasoning

The fuzzyDL reasoner, for a Twitter message, gives the fuzzy degree of membership to the class of relevant influenza messages. This result, along with the geolocation property of the tweet, is converted in the RDFs format. A RDF stream is made up of triples (subject, predicate, object), to which a timestamp is attached. The application generates information in the following format:

$(\prec: Tweet_i : hasInfluenza : val \succ, 2013-07-12T12:34:40)$
$(\prec: Tweet_i : hasLatitude : lat \succ, 2013-07-12T12:34:40)$
$(\prec: Tweet_i : hasLongitude : long \succ, 2013-07-12T12:34:40)$

In order to determine the number of influenza cases detected by the system over a period of 24 hours, the query in figure 4 is used.

In this example, we compute the number of tweets for the city of New York and its surroundings. This is accomplished by filtering the results according to the right geographical coordinates.

4 Experimental results and discussions

There are two important aspects in regard to the functionality of the system which need to be verified experimentally. First of all, the efficiency of the parsing method needs to be tested, in order to check the number of correctly categorised messages which enter the reasoning process. Table 3 presents the percentage of positively detected influenza messages, when a given word has been matched during the parsing algorithm.

Word found	Number of correctly categorised messages (in 100)
Flu	37%
Fever	48%
Cough, coughing	66%
Headache	69%
Sneeze, sneezing	33%

Table 3: Statistics of correctly categorised messages, by expression matched.

While cough and headache have a good percentage, other words have low detection efficiency. Hovewer, most of these are limitations of the currently used modules. For example, more than half of the incorrectly processed flu messages are introduced because the language used is not English. This is due to the fact that the Twitter Streaming API does not set the language property at the message body level, but at the user level. Therefore, a Twitter user can have the language set to English on his or her profile, yet write messages in other languages.

Moreover, as future improvements, the Stanford Dependencies parser can be used to filter common expressions which lead to incorrectly categorised messages. One such expression uses the word fever figuratively, in terms like $Bieberfever$ or $basketballfever$, found in more than 20 of the incorrectly detected tweets.

In order to test the functional aspect of the proposed solution, which is the detection of influenza rates, some experiments need to be performed. Over a course of one week (12-21 August), our application detected 8 influenza cases.

Unfortunately, it is difficult to compare these results with official reports. The New York State Department of Health monitors influenza rates only in the influenza season, which lasts from October to May. Therefore, there are no official weekly reports for August. However, from the archive of the Department of Health, for a period close to August, the week ending on June 30 2012[4], there were 13 laboratory confirmed cases of the influenza virus. The report for the week ending on the 11th of May 2013[5], 39 cases were laboratory confirmed.

Taking into consideration these results, it can be concluded that 8 cases for a week in August is a good approximate. However, for a more precise evaluation, the application needs to be tested in the winter period, when the influenza cases rise to the order of thousands per month. Moreover, the application has certain limitations which makes it difficult to simply compare the results with the official reports. For example, the application only receives 1% of the Twitter posts, and only processes messages which have the geolocation property set.

As future work, the application needs to be tested for larger periods of time. To properly compare results, the exact number of official cases should not be used. That is because there is no reason to believe that the officially reported number has to be the same as the number reported by the application. Instead, it should be seen if the fluctuation in official reports is also expressed in Twitter messages. For example, if the number of influenza cases rises by 10%, a similar trend should be reported by the application. Only after that, the solution can be said to correctly detect influenza epidemics, by reporting a number of cases over the epidemics threshold, in accordance to official reports (which are later generated). After that, the efficacy of the solution can be compared with the already existing ones, presented in the future section.

[4]http://www.health.ny.gov/diseases/communicable/influenza/surveillance/2011-2012/archive/2012-06-30/

[5]http://www.health.ny.gov/diseases/communicable/influenza/surveillance/2012-2013/archive/2013-05-11_flu_report.pdf

5 Related work

The most influent approach which makes use of the user generated data for detecting influenza epidemics is Google Flu Trends [1]. The data from five years of Google search archives and the official reports of CDC (Centers for Disease Control and Prevention) are used to create a machine learning model. The model can detect, based on the words used in user queries, influenza rates in a specific region. The result of applying this method, with a correlation between 0.85 and 0.96, is a good baseline for similar approaches.

Besides search engines, social networking platforms such as Twitter are becoming a very popular option for data provenance. Twitter messages are used for detecting influenza epidemics in [10]. The information is extracted using the Twitter Streaming API and, using a Support Vector Machine classifier, a tweet is labeled as negative (no useful information such as ambiguous or generic messages) and positive. The final correlation value of 0.89 is similar to the one in [1].

An influential work, based on the semantic analysis of Twitter messages is [11]. Linked Open Social Signals annotates tweets with semantic content and queries over them using the SPARQL language. The messages are extracted using the Twitter Streaming API, followed by a step of extracting relevant information. At the end of the extraction process, a micropost consists of the original text message, the author, the date of the post, its geolocation and a collection of entities, hashtags and URLs. This information is modelled under RDF(S)/OWL, by semantically annotating messages with concepts from the Linked Open Data Cloud. For example, the FOAF ontology (Friend of a Friend) is used to model Twitter users. The users have the possibility of querying the semantic information using SPARQL.

Another semantic approach to analyzing data is [12]. The data source used is Glue, a social network which allows its subscribers to connect with other users and share information regarding their favourite movies or sports. The platform makes use of semantic techniques for publishing topics in the form of RDF streams. C-SPARQL queries are used to reason over the data, using two methods of reasoning: deductive and inductive.

Twitter has been used in order to provide data regarding other public health issues, such as sentiment towards new tobacco products [13]. Again, the methods employed for detecting and classifying the relevant messages belong to the area of machine learning.

The research presented in this paper differs from the similar approaches presented in this section in two important aspects. One is the choice of semantic technologies for detecting influenza epidemics, instead of machine learning. The other one is the use of fuzzy semantic concepts, instead of just the classic OWL representation of the knowledge ontologies.

6 Conclusion

A novel approach to detecting influenza epidemics was presented. The fuzzy concepts employed help better shape the diagnosis protocol, by using linguistic variables appropriate to human language. Moreover, the semantic approach enables the solution to be further extended to concepts in the Symptom Ontology and to be useful in the broader context of the Semantic Web.

Acknowledgment: Part of this work was supported by the PN-II Romania-Moldova bilateral agreement ”ASDEC: Structural Argumentation for Decision Support with Normative Constraints”, 2013-2014.

References

[1] J. Ginsberg, M. Mohebbi, R. Patel, L. Brammer, M. Smolinski, and L. Brilliant, Detecting infuenza epidemics using search engine query data. *Nature*, vol. 457, no. 7232, pp. 1012-1014, 2009.

[2] E. D. Valle, S. Ceri, F. van Harmelen, and D. Fensel, It's a streaming world! reasoning upon rapidly changing information. *IEEE Intelligent Systems*, vol. 24, no. 6, pp. 83-89, 2009.

[3] M.-C. de Marneffe and C. D. Manning, The stanford typed dependencies representation. *Proceedings of the Workshop on Cross-Framework and Cross-Domain Parser Evaluation*, pp. 1-8, 2008.

[4] M. J. Garcia-Godoy, I. Navas-Delgado, and J. Aldana-Montes, Bioqueries: a social community sharing experiences while querying biological linked data. *Proceedings of the 4th International Workshop on Semantic Web Applications and Tools for the Life Sciences*, SWAT4LS 2011.

[5] D. F. Barbieri, D. Braga, S. Ceri, E. D. Valle, and M. Grossniklaus, Querying RDF streams with C-SPARQL. *SIGMOD Record*, vol. 39, no. 1, pp. 20-26, Sep. 2010.

[6] Bobillo, F., Straccia, U., fuzzyDL: An expressive fuzzy description logic reasoner. *2008 International Conference on Fuzzy Systems (FUZZ-08)*, pages 923-930.

[7] Pollock, J.L. Defeasible reasoning with variable degrees of justification. *Artif. Intelligence* 133(1-2), pp. 232-282, 2001.

[8] Groza A. and Letia, I.A. Plausible Description Logic Programs for Stream Reasoning. *Future Internet* 4(4), pp. 865-881, 2001.

[9] Arnold, M. D. Stefan Gravenstein, M. D. Michael Elliott, P. Michael Colopy, and M. D. Jo Schweinle. Clinical Signs and Symptoms Predicting Infuenza Infection. *Archives of International Medicine*, Vol. 160, no. No. 21, pp. 3243-3247, Nov. 2000.

[10] E. Aramaki, S. Maskawa, and M. Morita,Twitter catches the flu: detecting infuenza epidemics using twitter,*Proceedings of the Conference on Empirical Methods in Natural Language Processing*, ser. EMNLP '11, Association for Computational Linguistics, pp. 1568-1576, 2011.

[11] P. N. Mendes, A. Passant, P. Kapanipathi, and A. P. Sheth, Linked open social signals, *Proceedings of the 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, Volume 01, pp. 224-231, 2010.

[12] D. Barbieri, D. Braga, S. Ceri, E. D. Valle, Y. Huang, V. Tresp, A. Rettinger, and H. Wermser, Deductive and inductive stream reasoning for semantic social media analytics, *IEEE Intelligent Systems*, vol. 25, no. 6, pp. 32-41, 2010.

[13] Mark Mysln; Shu-Hong Zhu, Wendy Chapman, Mike Conway, Using Twitter to Examine Smoking Behavior and Perceptions of Emerging Tobacco Products, *Journal of Medical Internet Research*, 15(8):e174, 2013.

Radu Balaj
Technical University of Cluj-Napoca
Department of Computer Science
Intelligent Systems Group
Baritiu 28, RO-400391, Cluj-Napoca, Romania
E-mail: *radu.balaj@student.utcluj.ro*

Adrian Groza
Technical University of Cluj-Napoca
Department of Computer Science
Intelligent Systems Group
Baritiu 28, RO-400391, Cluj-Napoca, Romania
E-mail: *Adrian.Groza@cs.utcluj.ro*