Fact Validation Algorithm by Combination of Knowledge Graph and Neural Network

Technical Report · October 2019

DOI: 10.13140/RG.2.2.13495.27041

2 authors, including:

Michel Héon
Université du Québec à Montréal
49 PUBLICATIONS **157** CITATIONS

SEE PROFILE

Fact Validation Algorithm by Combination of Knowledge Graph and Neural Network

Michel Héon[1][0000-0001-7515-6382], Mehdi Camus[2]

[1] Université du Québec à Montréal, Québec, Canada
heon@cotechnoe.com
[2] MCA Industrial Solutions
mcamus@mca-industrial.com

Abstract. This article presents an algorithm combining semantic web technology with a neural network-based machine learning to validate the truth value associated with an RDF statement of the following structure: subject-predicate-object. The algorithm includes: 1) exploiting data semantics to generate new learning data, 2) using web data to enrich the characteristics of the learning vector, 3) exploiting statement semantics to guide the learning process. The article concludes with the presentation of preliminary results and a discussion about the algorithm, the results and future research perspectives.

Keywords: Ontology, Knowledge Graph, Neural Network, Semantic web, Fact validation, Machine Learning.

1 Introduction

Context. This research is carried out as part of the "Fact Validation Challenge"[1] of the International Semantic Web Conference 2019 which is divided into two tasks:

Task One: Fact Validation. Given a Resource Description Framework (RDF) statement about an entity (the correctness of a statement)

Task Two: Fact Validation at Scale. The evaluation of their scalability including runtime measurements and their ability to handle parallel requests.

The training dataset is made of 25K triplets evenly distributed according to the 5 following properties: `hasSameState, interactsWith, hasIndication, hasCommonIndication, hasCommonProducer`. As for the `'hasTruthValue'` property, it allows the true/false truth value to be associated with each statement.

Problematic and hypothesis. We want to develop a prediction filter unique to the two challenge spots. However, in addition to being effective at parallelized request, no ontological assumptions about the evaluation data are provided in the challenge statement. As a result, the prediction filter algorithm will instead have to be designed using a supervised machine learning approach of the type of a multi-layer neural network rather than by a logical inference approach.

Objective. Design a predictive fact validation filter of the truth value of an RDF statement using a supervised multi-layer neural network machine learning algorithm.

Related Work: Multiple works linking deeplearning and knowledge graph (KG) have inspired this research. We can note the work from Liu *et al*.[2] who propose to use semantic embedded in the KG in order to contextualize deeplearning while Deng *et al*.[3] use the semantics of the KG to enrich the training data of a neural network.

2 Prediction filter

The Fig.1 below shows the structure of the prediction filter used to estimate the truth value of an RDF statement. The prediction filter, which consists of several layers of data processing, is broken down into several processing pipelines containing a neural network specific to each dataset's property.

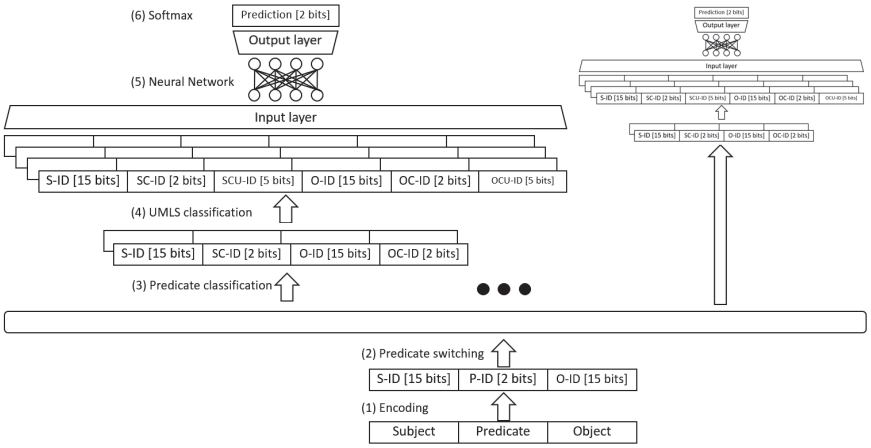

Fig. 1. Architecture of the truth value prediction filter of an RDF statement

The prediction filter breaks down according to the following hierarchy:

1. Encoding: This process is used to encode the RDF statement as a feature vector of the form: Subject-ID (S-ID) on 15 bits, Predicat-ID (P-ID) on 2 bits and Object-ID (O-ID) over 15 bits.
2. Predicate switching: At this stage, each vector is led to a treatment pipeline specific to the predicate ID number. The filter includes five pipelines, or one pipeline per predicate.
3. Predicate classification: this step has two objectives:
   a. Classify each subject (SC-ID) and object (OC-ID) of the statement as a 'Disease' or a 'Drugs'. The classification is achieved by adding a two-bit feature {0,1} for disease and {1, 0} for drugs.

b. Multiply the number of vectors depending on the nature of the predicate. The predicates `hasSameState`, `interactsWith`, `hasCommonIndication`, `hasCommonProducer` are symmetrical predicates. Thus, for these predicates, the value of truth remains unchanged regardless of the order of reading of the statement. For example: An A/p/B statement, which is true, will also be true if written in the form B/p/A. This step will increase the number of learning data from 25,000 statements to 45,000 statements.

4. UMLS Classification: The Unified Medical Language System (UMLS) [4] is a terminology directory of the biomedical field. This step consists of classifying each entity according to the UMLS terminology dictionary. The UMLS Classification has a twofold benefit to the filter. On one hand, it increases the granularity of the entity classification, and, on the other hand, it increases the number of data vectors since for a given entity there are sometimes several classifications in the UMLS. The SCU-ID and OCU-ID features, which are coded over 5 bits, respectively, represent the number associated with the semantic type (TUI) of the subject's AND of the object's of the statement to be processed.
5. Neural Network: This filter layer is a multilayer neural network. The number of hidden layers varies between 2 and 5 depending on the predicate selected (see Table 1). At its input, the network has 44 features and a Softmax output layer with two neurons. Each neuron is excited by the RELU-type activation function.
6. Softmax is the last layer of the filter. It presents the value of truth predicted by the filter. This is a value spread over two bits {0,1} to indicate that the statement is "true" and {1,0} to indicate a "false" value.

3 Results and Discussion

Neural Network Training: The training of the neural network is realized by the error backpropagation algorithm. The parameters (see Table 1): *Number of Hidden Layers, Number of Nodes by Layer and Learning Rate* are the values that condition the learning of each neural network which has been empirically optimized by the trial-and-error method inspired by the work of Guo *et al*.[5]. The variation in *batch size* is caused by the triplet numbers inferred at step 3 and step 4 of the filter algorithm (see Fig. 1). The *Minimun Error at Epoch* indicates the number of the epoch in which the Area Under Curve (AUC) is at a minimum.

Encoding and Transfer Function: Step 1 of encoding the algorithm appears to be decisive in the filter's performance. Various configurations have been experimented with, including replacing the limiting values {0, 1} with the values {-1, 1}, the use of vector standardization technique, or even, the use of the vectors converted in the Fourier space as the network's input. Other experiments on the choice of the neuron transfer function have been carried out with the functions: sigmoid and tanh. Our experiments conclude that the RELU function offers a better performance in the learning phase.

Vector generation tests: In order to avoid network overfitting, test data is submitted to the network during the training phase to assess its generalization capacity. For this research, 5% of the triplets were extracted for each of the training dataset predicates. For every triplet extracted, steps 1 to 4 of the algorithm (see Fig. 1) have been applied.

Performance Scoring: To date, the AUC filter performances 0.54 for task 1, 0.758 for task 2 and 2.69 ms of execution time for each prediction. The results show that the developed filter is operational for both tasks, which is a goal achieved only by our team. To improve the AUC of the filter, we think it would be necessary to refine the generation of test vectors notably by increasing the number of test triplets from 5% to 20% and by revising the vector encoding technique used in step 1 of the algorithm.

Table 1. Optimal values of neuron network parameters for each predicate

Predicate label	Number of Hidden Layers	Number of Nodes by Layer	Batch Size	Learning Rate	Minimun Error at Epoch	Average AUC
hasSameState	3	133	30387	0.02	4950	0,7759
interactsWith	4	177	37400	8.0E-4	5850	0,9886
hasIndication	4	45	8627	0.005	1050	0,8819
hasCommonIndication	5	89	34777	0.005	3050	0,8552
hasCommonProducer	8	89	35574	0.005	4400	0,9845

Conclusion

This research led to the development of a truth value prediction filter using an algorithm combining machine-learning neural networkand the use of logical inference mechanisms for classification of learning vectors and the automatic generation of triplets increasing the number of statements in the training dataset. Preliminary results demonstrate that further work needs to be done to consolidate the relevance of the approach used by our algorithm.

References

1. Ngomo, A.C.N., M. Röder, and Z.H. Syed. Semantic Web Challenge: Fact Validation Challenge. 2019 [cited 2019; Available from: https://iswc2019.semanticweb.org/challenges/.
2. Liu, Z. and X. Han, Deep Learning in Knowledge Graph, in Deep Learning in Natural Language Processing, L. Deng and Y. Liu, Editors. 2018, Springer Singapore. p. 117-145.
3. Deng, S., *et al*. Deep Learning for Knowledge-Driven Ontology Stream Prediction. 2019. Singapore: Springer Singapore.
4. Bodenreider, O., The unified medical language system (UMLS): integrating biomedical terminology. Nucleic acids research, 2004. 32(suppl_1): p. D267-D270.
5. Guo, C., *et al*. On calibration of modern neural networks. in Proceedings of the 34th International Conference on Machine Learning-Volume 70. 2017. JMLR. org.