Ontology population with deep learning-based NLP: a case study on the Biomolecular Network Ontology

Ali Ayadi, Ahmed Samet, François de Bertrand de Beuvron, Cecilia Zanni-Merk

Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 159 (2019) 572–581

Procedia Computer Science

www.elsevier.com/locate/procedia

23rd International Conference on Knowledge-Based and Intelligent Information & Engineering Systems

Ontology population with deep learning-based NLP: a case study on the Biomolecular Network Ontology

Ali Ayadi[a,1,*], Ahmed Samet[a], François de Bertrand de Beuvron[a], Cecilia Zanni-Merk[b]

[a] *ICUBE/SDC Team (UMR CNRS 7357)-Pole API BP 10413, Illkirch 67412, France*

[b] *LITIS Laboratory, Fédération CNRS Norm@STIC FR 3638, INSA de Rouen Normandie, Avenue de l'Université, 76801 Saint-Etienne-du Rouvray, France*

Abstract

As a scientific discipline, systems biology aims to build models of biological systems and processes through the computer analysis of a large amount of experimental data describing the behaviour of whole cells. It is within this context that we already developed the Biomolecular Network Ontology especially for the semantic understanding of the behaviour of complex biomolecular networks and their transittability. However, the challenge now is how to automatically populate it from a variety of biological documents. To this end, the target of this paper is to propose a new approach to automatically populate the Biomolecular Network Ontology and take advantage of the vast amount of biological knowledge expressed in heterogeneous unstructured data about complex biomolecular networks. Indeed, we have recently observed the emergence of deep learning techniques that provide significant and rapid progress in several domains, particularly in the process of deriving high-quality information from text. Despite its significant progress in recent years, deep learning is still not commonly used to populate ontologies. In this paper, we present a deep learning-based NLP ontology population system to populate the Biomolecular Network Ontology. Its originality is to jointly exploit deep learning and natural language processing techniques to identify, extract and classify new instances referring to the BNO ontology's concepts from textual data. The preliminary results highlight the efficiency of our proposal for ontology population.

Keywords: Ontology population; Knowledge acquisition; Natural language processing; Deep learning; Biomolecular Network Ontology.

1. Introduction

Complex biomolecular networks include a series of networked complex systems ranging from genomic and transcriptomic to proteomic and metabolomic ones [1]. For studying these complex biomolecular systems, we represent them as networks in which the nodes represent the entities of the complex system (genes, proteins, metabolites, etc.),

* Corresponding author. Tel.: +33 6 56 76 34 46.
E-mail address: ali.ayadi@unistra.fr

1877-0509
10.1016/j.procs.2019.09.212

and the edges represent the possible interactions among them (physical interactions or chemical transformations, etc.). These complex networks are considered as systems that dynamically evolve from a state to another so that the cell can adapt itself to changes in its environment [2]. This issue has already been addressed in our previous works [2], where we develop an ontology, the Biomolecular Network Ontology (BNO)[1], for modeling all the necessary biological knowledge to study and reason on complex biomolecular networks.

The BNO ontology has been built manually and under expert guidance, a process known as ontology learning. Indeed, we have already defined the concepts and relations of the BNO ontology, which represent the TBox of the ontology. This TBox consists of twenty-five classes and twelve properties. However, what we really need now is how to instantiate the BNO ontology from biological documents. In other words, how to populate the BNO ontology automatically (enrich its ABox). This process is called ontology population.

Moreover, with recent advances in high throughput biology techniques, intense research in molecular biology has led to major discoveries in cellular components, producing an important volume of knowledge about these components [2]. This biological documents can be considered as a vital source of knowledge for understanding the behaviour of complex biomolecular networks. It would, therefore, be helpful to exploit this 'omic' knowledge to populate our ontology and more contribute to the understanding of the behaviour of complex biomolecular networks. However, this process is greatly dependent on the knowledge captured in the documents by the expert and professional biologists. Manual processing of biological documents is extremely expensive in time and resources, and prone to human error.

This paper investigates the problem of ontology population by proposing an ontology population system for knowledge acquisition from textual 'omic' resources, and automatically populate the BNO ontology. This system aims to identify and extract useful textual terms and assign them with respect to the predefined concepts (classes), instances (individuals), attributes (data properties) and relationships (object properties) of the BNO ontology. Our proposed ontology population system combines NLP techniques and deep learning. The originality of our proposal is to jointly exploit deep learning and natural language processing techniques to identify, extract and integrate new instances that populate our ontology from textual data. The preliminary results highlight the efficiency of our proposal for ontology population. Indeed, we have recently observed the emergence of deep learning techniques that provide significant and rapid progress in several domains, in particular in the process of deriving high-quality information from text. Despite its significant progress in recent years, deep learning is still not commonly used in the ontology population process.

The remaining of this paper is organised as follows. In Section 2 we present a brief introduction to existing studies that tackle the same problem but with different approaches. Then, we describe our proposed approach in Section 3. In section 4, we compare the obtained results with the proposed approach to those obtained with the user-centric method and evaluates its efficiency. Finally, we provide the conclusion and prospects of our work in Section 5.

2. Related work

The purpose of this section is to provide a thorough and comprehensive overview of existing works that address the ontology population process.

2.1. Acquiring contextual information

Knowledge acquisition from raw input resources is required by ontology enrichment and ontology population. According to Ksiksi A. and Amiri H. [3] and Harb et al. [4], we distinguish two types of methods for extracting domain specific terms, concepts and associations among them, *linguistic techniques* and *statistical techniques*. The first type considers the hypothesis that the grammatical structure reflects semantic dependencies. They aim to determine the semantic dependencies of the terms within a sentence. They use the grammatical function of a word within a sentence. These methods have the capacity to extract terms and the relations among them [4]. The second type identifies the terms according to their distribution in the texts (using mutual information, tf-idf measurements). These statistical techniques are provided from data mining, machine learning and information retrieval. These techniques have the capacity to identify new candidate terms for the ontology population and enrichment, but cannot place them in the ontology, without a tiresome human intervention [3].

[1] https://github.com/AliAyadi/BNO-ontology-version-1.0

2.2. Ontology population

Ontology population from texts has been widely used in the community of knowledge engineering. According to a comprehensive study in [5], we can distinguish three main categories of ontology population systems: (i) *rule-based ontology population systems*, (ii) *ontology population systems that use machine learning*, and (iii) *ontology population systems that use statistical approaches*. This classification is based on the techniques used to perform the task of extracting domain specific terms presented above.

-*Rule-based ontology population systems*: The rule-based ontology population systems use lexico-syntactic patterns to locate concept and relation instances in the text. Indeed, they use a set of rules (e.g. syntactic, grammatical, orthographic features) in combination with a list of dictionaries (e.g. the list of genes, cells, etc.) that are manually predefined by experts. In the era of systems biology, we can cite the works of Finkelstein M. et al. [6], Yangarber et al. [7], Ibrahim et al. [8], Harith et al. [9], Makki et al. [10], Ananiadou et al. [11], Ravikumar et al. [12], and Eftimov et al. [13]. However, these methods require manual effort to build the extraction rules and validate the patterns. Moreover, the extracted patterns can be incorrect because of the lack of deep linguistic analysis. These semi-automatic systems do not provide solutions for consistency problems.

-*Ontology population systems that use statistical approaches*: This kind of ontology population systems uses statistical approaches, such as the works of Yoon et al. [14], Maynard et al. [15], and Tanev et al. [16]. They are based on semantic similarity measurements and fitness functions for computing the textual similarity between the extracted terms and instances in the ontology. However, these approaches are not appropriate for all situations, for example, they cannot treat terms not covered by synonym dictionaries or are not suitable for understanding abbreviations, etc.

-*Machine learning ontology population systems*: These machine learning ontology population systems employ a classification model to identify candidate instances (supervised or unsupervised). They use machine learning algorithms to extract instances from unstructured text. Most of these approaches are based on the Learning Pattern by Language Processing (LP^2) algorithm for supervised learning based Support Vector Machine or on Lazy-NLP and First Order Inductive Learning (FOIL) algorithms. Among these works, Celjuska et al. [17], Etzioni et al. [18], Chun et al. [19], Jiang et al. [20], and Souili et al. [21].

-*Deep learning ontology population systems*: Only a few deep learning ontology population systems have been proposed in the literature. Most of them are based on some form of recurrent neural networks (RNN), such the works of Zeng et al. [22], Chen et al. [23], and Liu et al. [24]. These systems are domain-dependent and require domain-specific tagged resources. However, in these techniques, no expert intervention is required and none of them performs consistency or redundancy checks. These systems are extensively discussed in [5].

There are also some other hybrid approaches of ontology population systems, such as those proposed by Torri et al. [25] who combine both rule-based and ML methods, have shown good performance on gene-name recognition tasks, and the works of Specia et al. [26] which employ knowledge-based and corpus-based methods.

3. Proposed methodology

3.1. Architecture of the proposed methodology

Figure 1 shows a schematic representation of the steps involved in the proposed ontology population system and the diverse methods used in each step. The initial step describes the input of the proposed ontology population system which consists of two major components, (i) the set of biological documents representing the knowledge resources and (ii) the Biomolecular Network Ontology which is a domain ontology for describing the behaviour of complex biomolecular networks. The second step represents the knowledge extraction process which aims to identify the useful knowledge from the input biological documents and extract the candidate instances of concept, relations and attributes using the join help of the BNO ontology and the deep learning-based NLP techniques to successfully perform the extraction task. Finally, the last step aims to verify the redundancy and consistency of the extracted instances in relation to the existing stored knowledge in the BNO ontology. This step is ensured through expert intervention. Then, the filtered instances migrate to populate the BNO ontology. These steps are discussed in the following sections.

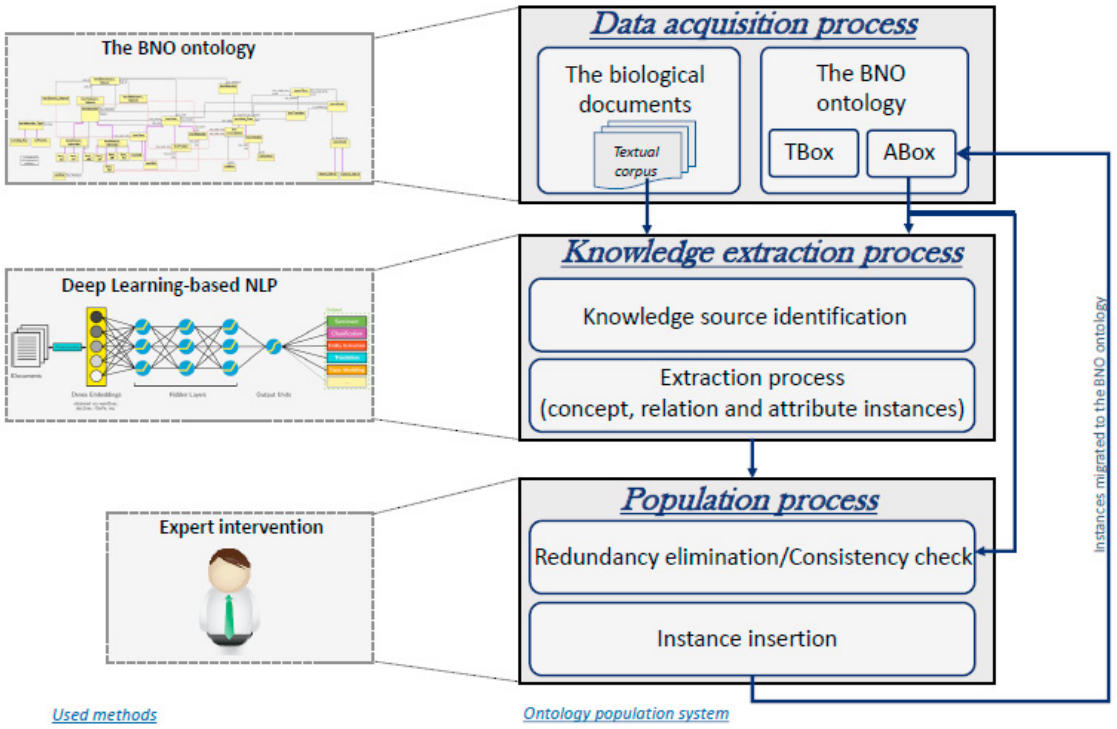

Fig. 1. Architecture of the proposed methodology.

3.2. The data acquisition process

This first step consists of searching web documents and local files related to the domain of our ontology. As our goal is to populate the BNO ontology, we use biological documents related to complex biomolecular networks. This initial step aims to prepare the biological documents for processing by the next phase, the preprocessing phase using NLP techniques. Thus, the two main components of this step are the Biomolecular Network Ontology which contains instances in each concept, and a set of heterogeneous documents related to the topic of our domain ontology.

3.3. The knowledge extraction process

The knowledge extraction process consists of two main steps, the preprocessing data to analyse and identify the knowledge source, and the extraction process for the classification and of the candidate terms to the ontology population task. These steps are ensured by the natural language techniques.

3.3.1. Text preprocessing

The preprocessing step aims to transform raw data into an understandable format. In our context, this step aims to make the input biological documents easier to work with and present them into a form that is more predictable and analysable for the next deep learning task. In this step, we use basic natural language processing (NLP) tasks, such as, (i) the *tokenization* which covers the text segmentation and its lexical analysis. This task aims to split longer strings of text into smaller pieces. Larger paragraphs of text are converted into sentences, sentences are also converted into words, etc. This task relies on two pre-trained algorithms, the Punkt sentence tokenizer and Penn Treebank word tokenizer from NLTK[2]. And (ii) the *normalisation* which is an important task consisting of a series of related tasks

[2] https://www.kaggle.com/nltkdata/punkt

for converting all text to the same case (lower or upper), removing punctuation, converting numbers to their word equivalents, removing the general stop words (”the”, ”a”, etc.), etc. This task is done using a simple technique, the Min-Max normalization, which allows to specifically fit the data in a pre-defined boundary. It is often known as feature scaling where the values of a numeric range of a feature of data (a property) are reduced to a scale between 0 and 1.

3.3.2. Deep learning step

To accomplish this step, we draw inspiration from the works of Albukhita et al. [27]. The preprocessing textual data are processed to produce a language model based on word embedding for all provided corpus. We use the Word2vec which is a popular algorithm to learn word embeddings using a shallow neural network. This technique allows representing words as vectors, usually in a space of a few hundred dimensions. The vector representing each word is obtained through an iterative algorithm starting from a large amount of text. The algorithm tries to place the vectors in space in order to approximate semantically close words and to move away semantically distant words.

The Word2vec algorithm [28] can be described as follows: The first phase of Word2vec is to associate each word w of the prepared data with a randomly initialised vector denoted by $v_w \in \mathbb{R}^n$. For each word w within each sentence of the training corpus, a window of words called *context* and denoted by c around the word w is considered. In our case, the size of this context is around 5 to 10 words. We define $P(D = 1|c, w)$ the probability that a word c is in the context of another word w. This probability $P(D = 1|c, w)$ is defined as the sigmoid[3] of the scalar product of the vectors of the two words as defined by equation (1).

$$P(D = 1|c, w) = \frac{1}{1 + e^{v_c . v_w}} \tag{1}$$

Conversely, we define $P(D = 0|c, w)$, the probability that a word c do not belong to the context of another word w. $P(D = 0|c, w)$ is defined by equation (2).

$$P(D = 1|c, w) = 1 - P(D = 0|c, w) \tag{2}$$

Then, we define the optimization objective function $L(V)$ using equation (3).

$$L(V) = arg\ max \sum_{(w,c)\in A} \log P(D = 1|c, w) + \sum_{(w,c^{'})\in B} P(D = 0|c^{'}, w) \tag{3}$$

V denotes the set of all vectors v of the words w that describe our model and for which we are trying to find the optimal values in order to maximise the objective function $L(V)$. We have used the stochastic gradient descent which is the dominant method used to train deep learning models. This simple optimization procedure works by having the model make predictions on training data and using the error on the predictions to update the model in such a way as to reduce the error. The goal of the algorithm is to find model parameters (coefficients) that minimise the error of the model on the training dataset. It does this by making changes to the model that move it along a gradient or slope of errors down toward a minimum error value. This gradient descent approach is used to determine the optimal values of all the vectors v corresponding to the words w. More details about this gradient descent approach can be found in this post[4]. This algorithm provides vectors that bring together words that are semantically close (’are embedded nearby each other’) and move away from words that are semantically distant. Indeed, semantically related words are frequently in the same context. Consequently, these vectors model the word embeddings. To do this, we used

[3] https://en.wikipedia.org/wiki/Sigmoid_function

[4] https://machinelearningmastery.com/gradient-descent-for-machine-learning/

the Python open source *Gensim*[5] library for the implementation of the Word2vec algorithm. More details about this algorithm can be found in [28].

The candidate words embedding are obtained with the trained Word2vec algorithm are compared with the concept classes and properties of the BNO ontology. For each ontology terms (concept, property or attribute), a number of candidate vectors are computed using the vector embedding vectors of the instances. Indeed, this Word2vec is able to model and provide the relationship between predictive variables (input words) and a target variable (the target ontological terms). The recognition consists in computing the probability of each group of words and automatically aggregate it according to the target ontological terms. In other words, the seed concepts of the BNO ontology are used to organise the results of the Word2vec algorithm. This method considers also the inter-class similarities.

3.4. Ontology population process

The ontology population process has two main phases. The first one consists of the expert ontologist interventions for checking and controlling the redundancy and consistency of the suitable words embedding from the precedent deep learning-based NLP technique. During this phase, the ontologist evaluates the candidate terms and expresses his intention to modify the resulting propositions by applying some corrections (accept, reject, move, delete, create, split, merge, group, etc.). This task contributes to training our deep learning based-NLP method to better adapt its results to the ontologist's feedback. The second phase is the insertion of the new terms in their appropriate location in the domain ontology. This step consists of placing the candidate terms while preserving the coherence of the pre-established concepts and relations in the BNO ontology. This step is also based on the results provided by the above deep learning-based NLP technique. Indeed, the insertion of the new terms respects the classification done by the deep learning-based NLP technique. To perform this task, we use the Jean-Baptiste Lamy's packages [29]. These packages, *Owlready* and $Python - skos$, provide a large variety of methods for treating ontologies, in particular for the insertion of instances in the ontology.

4. Preliminary results

The experiment process is done by following the steps of the proposed approach. The first step is preparation for a set of textual documents related to complex biomolecular networks. The second one is the preparation of these biological texts and generate an instance classification. Then, the last one is to populate the BNO ontology and evaluate our proposed approach. It is important to note that the results presented in this section are preliminary results since the study is still underway.

Training and test data. To prepare a test set for checking our approach, we have adopted a small corpus consisting of 15 textual documents which have the characteristics needed for our work. These documents belong to the PubMed Central[6] (PMC), a free full-text archive of biomedical and life sciences journal literature. Indeed, the PubMed Central archive covers a large number of articles treating complex biomolecular networks. By doing the first search, i.e. typing "transittabilitty complex biomolecular networks", we obtain 16301 articles. Among them, we only select 15 articles in order to facilitate the manual selection of instances. This test set of 15 articles and the BNO ontology containing 18 instances represent the input of our approach. Furthermore, we treat also this small corpus so that humans can easily treat it by hand. We treat these different articles in order to obtain a set of structured information related to the domain of the transittability of complex biomolecular networks. Indeed, sentences in selected biological papers are sometimes long and may contain more than 400 words. Moreover, not all the information contained in these articles are useful for us. Some information may also be duplicated. Therefore, we did not select all the papers, but we treat them to only extract relevant sections related to our ontology topic necessary. Table 1 shows an extract of the training corpus. For the sake of space, we did not cite the articles' references.

[5] https://radimrehurek.com/gensim/about.html
[6] https://www.ncbi.nlm.nih.gov/pmc/

Table 1. The training corpus. Given are the title of the selected article, its authors, its printing year, and the number of retained words for training and evaluation.

Title	*Authors*	*Year*	*#words*
Mining and state-space modeling and verification of sub-networks from large ...	Hu et al.	2007	108
Discovery of a kernel for controlling biomolecular regulatory networks	Kim et al.	2013	92
Transittability of complex networks and its applications to regulatory ...	Wu et al.	2014	65
Big biological data: challenges and opportunities	Li et al.	2014	66
Systems proteomics view of the endogenous human claudin protein family	Liu et al.	2016	91
Stochastic simulation of biomolecular networks in dynamic environments	Voliotis et al.	2016	50
Physical controllability of complex networks	Wang et al.	2017	98
Energy-based analysis of biomolecular pathways	Gawthrop et al.	2017	80
Bipartite graphs in systems biology and medicine: a survey of methods ...	Pavlopoulos et al.	2018	123
The phenotype control kernel of a biomolecular regulatory network	Choo et al.	2018	75
Modular DNA strand-displacement controllers for directing material expansion	Fern et al.	2018	100
BioNet. A Python interface to NEURON for modeling large-scale networks	Gratiy et al.	2018	114
Genomic data integration systematically biases interactome mapping	Skinnider et al.	2018	85
MatrixDB: integration of new data with a focus on glycosaminoglycan interactions	Clerc et al.	2018	36
Plausible Emergence of Autocatalytic Cycles under Prebiotic Conditions	Piotto et al.	2019	67

Experiments. After selecting the test set, we implement our approach for identifying the candidate terms (future instances) from this corpus to populate the BNO ontology. The proposed approach was implemented in Python using the Natural Language Toolkit[7] (NLTK), and the *Pandas*, *Matplotlib*, *Seaborn* and *Numpy* packages for performing the preprocessing and NLP tasks. The Word2vec algorithm is coded also in Python and is a part of the *gensim* package. Our experiment was performed on a computer equipped by a processor Intel Core i5-4460 CPU @ 3.20GHz $\times$ 4 and a 15,5 Gb main memory. For training the SKIP-G model of the proposed deep learned-based NLP approach, we set the parameters as follows: the vector size is fixed at 250 for modeling the size of the generated vectors, the window is fixed at 7 words for representing the context around the keyword in the training model, the sample size is fixed to 10^{-3} which is a threshold for occurrence of words.

To evaluate this proposed approach, we compare it with a user-centric ontology population that includes human at both processes of "*Identification of instances*" and "*Classification of instances*" with the same corpus. Therefore, we manually identified the candidate instances within the training corpus and also manually classify the obtained candidate instances according to the BNO ontology concepts. The number of identified instances is 1136 instances. All these steps have been performed by hand and with the assistance of an expert. Consequently, we divided our evaluation into two steps, the evaluation of the "Identification of instances" and the evaluation of the "Classification of instances". The effectiveness of both processes is evaluated by comparing the obtained results of our proposed approach with those generated manually with the user-centric method.

The measures of Precision and Recall from the information retrieval domain are used for performance evaluation [30] considering the number of instances rightly classified. Precision measures the ratio between the number of instances correctly identified and the number of instances identified, in the process of "*Identification of instances*", and measures the ratio between the number of instances correctly classified and the number of instances classified in the second process of "*Classification of instances*". We compute it using equation (4).

$$Precision = \frac{Number\ of\ candidate\ instance\ correctly\ identified\ /\ classified}{Number\ of\ instance\ identified\ /\ classified} \tag{4}$$

[7] http://www.nltk.org/

Then, we use the Recall to measure the ratio between the number of instances correctly identified and the number of instances in the corpus, in the process of "*Identification of instances*", and the ratio between the number of instances correctly classified and the number of instances in the corpus,in the second process of "*Classification of a instances*". The Recall of the system is computed using equation (5).

$$Recall = \frac{Number\ of\ candidate\ instance\ correctly\ identified\ /\ classified}{Number\ of\ instance\ identified\ in\ the\ corpus} \tag{5}$$

Using equation (6), we compute the F-measure of both processes. This metric gives an harmonic mean of the Precision and Measure metrics.

$$F - measure = \frac{(2 * Precision * Recall)}{(Precision + Recall)} \tag{6}$$

Evaluation and validation. The goal of this evaluation is to show the capabilities of the proposed approach to correctly identify and classify the candidate instances. To do this, we follow the evaluation of the ontology population from Faria et al. [30]. This evaluation consists of computing the *Precision* and *Recall* measures of our proposed approach according to the user-centric experiment, for both processes of "*Identification of instances*" and "*Classification of instances*" using the same corpus.

1 - Identification of instances: Table 2 illustrates the results of the evaluation of the "Identification of instances process" of our proposed approach corresponding to the first line of the Table (Automatic identification) with respect to the user-centric ontology population corresponding to the second line of the Table (Manual identification). The first row of the table shows the performance of instances identified by the proposed approach: the terms identified as instances are 960, those correctly identified are 656 from the 1250 instances in the corpus. These values correspond to a precision of 68.33%, a recall of 52.48% and an F-measure of 59.36%. The second experiment is conducted manually, the terms identified as instances are 1136, those correctly identified are 710 from the 1250 instances in the corpus. These values correspond to a precision of 62.50%, a recall of 56.80% and an F-measure of 59.51%.

Table 2. Results of the evaluation of the identification of candidate instances process.

Task	*Inst. in the corpus*	*Inst. identified*	*Inst. correctly identified*	*Precision*	*Recall*	*F-measure*
Automatic identification	1250	960	656	68.33%	52.48%	59.36%
Manual identification	1250	1136	710	62.50%	56.80%	59.51%

2 - Classification of candidate instances: Table 3 summarises the results of the evaluation of the Classification of candidate instances process. Manually, the number of instances classified are 550, those correctly classified are 500 from the 710 instances in the corpus (corresponding to the instances correctly identified from the first process). These values correspond to a precision of 90.90%, a recall of 70.42% and an F-measure of 79.36%. In the second experiment corresponding to our proposed approach, the number of instances classified are 484, those correctly classified are 435 from the 656 instances in the corpus (corresponding to the instances correctly identified from the first process). These values correspond to a precision of 89.87%, a recall of 66.31% and an F-measure of 76.31%.

The analysis of document contents in the testing corpus showed that it contains a lot of "unusable words", words that are often repeated and do not have important weight but have an adverse impact on our classification approach. Table 4 presents some examples of instances classified incorrectly by the proposed approach and their key concepts. For example, the instance "TH" is interpreted by the proposed approach as a protein, however by definition the "TH"[8]

[8] https://ghr.nlm.nih.gov/gene/TH

is a gene that codes for the enzyme tyrosine hydroxylase. In this case, the proposed approach confounded tyrosine hydroxylase gene abbreviated as "TH" by tyrosine hydroxylase enzyme abbreviated as "TH-HZ" considering it as a protein rather than a gene. Another example is the "Cdx", which is manually classified as a transcription factor, is neither recognised as "Protein" or "Transcription_factor". This is due to the "vague" and fuzzy properties of the candidate instance. The proposed approach cannot classify this instance because it is a particular case of protein family having some properties very close to both types of classes ("Protein" and "Transcription_factor") and therefore it is not possible for our approach to distinguish between them. This is due to the fuzzy definition of the properties of the classes. To avoid these cases, additional properties have to be included or existing properties have to be omitted.

Table 3. Results of the evaluation of the classification of instances process.

Task	*Inst. in the corpus*	*Inst. classified*	*Inst. correctly classified*	*Precision*	*Recall*	*F-measure*
Manual classification	710	550	500	90.90%	70.42%	79.36%
Automatic classification	656	484	435	89.87%	66.31%	76.31%

Table 4. Examples of instances missclassified by the proposed approach: the instance identifier, its predicted concept and its key concept corresponding to its correct BNO class.

Instance ID	*instance*	*Predicted concept*	*Key concept*
11	TH	Protein	Gene
25	Cdx	—	Transcription_factor
67	DNA_damage	DNA	Stimuli
71	oncogene	Gene	—

Based on the Precision, Recall and F-measure metrics, we can note that these first results are encouraging. Indeed, the approach we have proposed has been defined to facilitate the automatic population of the BNO ontology, while avoiding the manual efforts of document annotation and, instances identification and extraction. The proposed method can perform the population process from heterogeneous unstructured documents, and without manually tagging or annotating the documents by hand. This proposed approach automatically identified and classified new instances to populate the BNO ontology. These instances are coherent and were also validated by expert biologists. The proposed method may allow treating large corpus so we can benefit from measures of semantic relatedness. However, we need to test our approach using a more large dataset to enhance the performance and quality of the proposed approach. Indeed, choosing a large number of biological documents may impact the proposed approach performance.

5. Conclusions and future work

Diverse ontology population systems have been proposed in the literature. Their limitations come from the fact that they cannot perform the population process from heterogeneous unstructured documents, and without manually tagging or annotating the documents by hand. However, the annotation is usually time-consuming and therefore generally expensive. In this paper, we presented a deep learning-based NLP method for ontology population from biological texts and apply it to instantiate the Biomolecular Network Ontology. The originality of our approach is that it mutually exploits the expressibility and trainability of deep learning and natural language processing techniques to identify, extract, classify and integrate new concepts and specialisations of relationships to enrich the BNO ontology from textual data. So, in contrast to traditional NLP methods which focus on the syntactic representation, the contribution of deep learning allows them to focus on semantic representation which enables him to distinguish certain contexts.

Our current work focuses on implementing a system prototype for providing the user with sophisticated interfaces to simplify the interaction among the different approach modules. These interfaces will facilitate the selection of corpus, the choice of the appropriate preprocessing techniques and the setting of the deep learning parameters. Our future work aims at improving the deep learning-based NLP approach in order to obtain more performance.

References

[1] Estrada, Ernesto. (2012) "Complex biomolecular networks: challenges and opportunities." *Briefings in Functional Genomics* **11)** (6): 417–419.

[2] Ayadi, Ali, Zanni-Merk, Cecilia, de Beuvron, François De Beuvron, Thompson, Julie, and Krichen, Saoussen. (2019) "BNO—An ontology for understanding the transittability of complex biomolecular networks." *Journal of Web Semantics*.

[3] Ksiksi, Asma, and Hamid Amiri. (2018) "Using Association Rules to Enrich Arabic Ontology." *Engineering, Technology and Applied Science Research*. **8(3)** (3): 2914–2918.

[4] Harb, Ali, Kafil Hajlaoui, and Xavier Boucher. (2011) "Competence mining for collaborative virtual enterprise." *In : Working Conference on Virtual Enterprises*, 351–358.

[5] Liang, Hong, Sun, X., Sun, Xiao, and Gao, Yunlei. (2017) "Text feature extraction based on deep learning: a review." *EURASIP journal on wireless communications and networking*. ; **2017** (1):211.

[6] Finkelstein-Landau, Michal, and Morin, Emmanuel. (1999) "Extracting semantic relationships between terms: Supervised vs. unsupervised methods." *In: International Workshop on Ontological Engineering on the Global Information Infrastructure*, 71–80.

[7] Yangarber, Roman, and Grishman, Ralph. (1998) "Description of the Proteus/PET system as used for MUC-7 ST." *In: Seventh Message Understanding Conference (MUC-7): Proceedings of a Conference Held in Fairfax, Virginia* **1998**.

[8] Ibrahim, Zaharudin, Noah, Shahrul Azman, and Noor, Mahanem Mat. (2010) "Rules for ontology population from text of Malaysia medicinal herbs domain." *In: International Conference on Rough Sets and Knowledge Technology. Springer*, Berlin, Heidelberg, 386–394.

[9] Harith, Alani, Kim, Sanghee, Millard, David E., Weal, Mark J., Hall, Wendy, Lewis, Paul H., and Shadbolt, Nigel R. (2003) "Automatic ontology-based knowledge extraction and tailored biography generation from the web." *IEEE Intelligent Systems* **18** (1): 14–21.

[10] Makki, Jawad, Alquier, Anne-Marie, and Prince, Violaine. (2009) "Ontology population via NLP techniques in risk management." *International Journal of Humanities and Social Science (IJHSS)* **3** (3): 212–217.

[11] Ananiadou, Sophia, Pyysalo, Sampo, Tsujii, Jun'ich, and Kell, D. B. (2010) "Event extraction for systems biology by text mining the literature." *Trends in biotechnology* **28** (7): 381–390.

[12] Ravikumar, K. E., Wagholikar, K. B., and Liu, H. (2014) "Towards pathway curation through literature mining–a case study using PharmGKB." *In: Biocomputing* **2014**: 352–363.

[13] Eftimov, Tome, Seljak, Barbara Koroušić, and Korošec, Peter. (2017) "A rule-based named-entity recognition method for knowledge extraction of evidence-based dietary recommendations." *PloS one* **12** (6):0179488.

[14] Yoon, Hee-Geun, Han, Yong Jin, Park, Seong-Bae, and Park,Se-Young. (2007) "Ontology population from unstructured and semi-structured texts." *In: Sixth International Conference on Advanced Language Processing and Web Information Technology (ALPIT 2007). IEEE*, 135–139.

[15] Maynard, Diana, Li, Yaoyong, and Peters, Wim. (2008) "NLP Techniques for Term Extraction and Ontology Population."

[16] Tanev, Hristo, and Magnini, Bernardo. (2006) "Weakly supervised approaches for ontology population." *In: 11th Conference of the European Chapter of the Association for Computational Linguistics*.

[17] Celjuska, David, and Vargas-Vera, Maria. (2004) "Ontosophie: A semi-automatic system for ontology population from text." *In: International Conference on Natural Language Processing (ICON)*, 60.

[18] Etzioni, Oren, Cafarella, Michael, Downey, Doug, Popescu, A. M., Shaked, T., Soderland, S., ... and Yates, A. (2005) "Unsupervised named-entity extraction from the web: An experimental study." *Artificial intelligence* **165** (1): 91–134.

[19] Chun, CHUN, Hong-Woo, Tsuruoka, Yoshimasa, Kim, Jin-Dong, Shiba, R., Nagata, N., Hishiki, T., and Tsujii, J. I. (2006) "Extraction of gene-disease relations from Medline using domain dictionaries and machine learning." *In: Biocomputing 2006*, 4–15.

[20] Jiang, M., Chen, Y., Liu, M., Rosenbloom, S. T., Mani, S., Denny, J. C., and Xu, H. (2011) "A study of machine-learning-based approaches to extract clinical entities and their assertions from summaries." *Journal of the American Medical Informatics Association* **18** (5): 601–606.

[21] Souili, Achille, Cavalucci, Denis, et Rousselot, François. (2015) "Natural Language Processing (NLP)–A Solution for Knowledge Extraction from Patent Unstructured Data." *Procedia engineering* **131**: 635–643.

[22] Zeng, Daojian, Liu, Kang, Lai, Siwei, Zhou, G., and Zhao, J. (2014) "Relation classification via convolutional deep neural network."

[23] Chen, Yu, Li, Wenjie, Liu, Yan, Zheng, D., and Zhao, T. (2010) "Exploring deep belief network for chinese relation extraction." *In: CIPS-SIGHAN Joint Conference on Chinese Language Processing*.

[24] Liu, ChunYang, Sun, WenBo, Chao, WenHan, and Che, W. (2013) "Convolution neural network for relation extraction." *In: International Conference on Advanced Data Mining and Applications. Springer* , Berlin, Heidelberg, 231–242.

[25] Torii, Manabu, Hu, Zhangzhi, Wu, Cathy H., and Liu, H. (2009) "BioTagger-GM: a gene/protein name recognition system." *Journal of the American Medical Informatics Association* **16** (2): 247–255.

[26] Specia, Lucia, and Motta, Enrico. (2006) "A hybrid approach for extracting semantic relations from texts." *In: Proceedings of the 2nd Workshop on Ontology Learning and Population: Bridging the Gap between Text and Knowledge*, 57–64.

[27] Albukhitan, Saeed, Helmy, Tarek, and Alnazer, Ahmed. (2017) "Arabic ontology learning using deep learning." *In: Proceedings of the International Conference on Web Intelligence. ACM*, 1138–1142.

[28] Mikolov, Tomas, Sutskever, Ilya, Chen, Kai, Corrado, G. S., and Dean, J. (2013) "Distributed representations of words and phrases and their compositionality." *In: Advances in neural information processing systems*, 3111–3119.

[29] Lamy, Jean-Baptiste. (2017) "Owlready: Ontology-oriented programming in Python with automatic classification and high level constructs for biomedical ontologies." *Artificial intelligence in medicine*. **80**, 11–28.

[30] Faria, Carla, Serra, Ivo, and Girardi, Rosario. (2014) "A domain-independent process for automatic ontology population from text." *Science of Computer Programming* **95**: 26–43.