Automatic Construction of Domain Ontology Using Wikipedia and Enhancing it by Google Search Engine

Sedigheh Khalatbari
Department of Computer Engineering, University of Guilan, Rasht, Iran
khalatbari@msc.guilan.ac.ir
Seyed Abolghasem Mirroshandel*
Department of Computer Engineering, University of Guilan, Rasht, Iran
asedghasem@yahoo.com

Received: 04/May/2015 Revised: 20/Nov/2015 Accepted: 08/Dec/2015

Abstract

Information and resources available on the Web are growing increasingly and web users need to have a common understanding of them. The Semantic Web whose most important role is to help machine to understand and analyze the existing data on the Web, has not been used commonly, yet. The foundation of the Semantic Web are ontologies. Ontologies play the main role in the exchange of information and development of the Lexical Web to the Semantic Web. Manual construction of ontologies is time-consuming, expensive, and dependent on the knowledge of domain engineers. Also, Ontologies that have been extracted automatically from corpus on the Web might have incomplete information. The main objective of this study is describing a method to improve and expand the information of the ontologies. Therefore, this study first discusses the automatic construction of prototype ontology in animals' domain from Wikipedia and then a method is presented to improve the built ontology. The proposed method of improving ontology expands ontology concepts through Bootstrapping methods using a set of concepts and relations in initial ontology and with the help of the Google search engine. A confidence measure was considered to choose the best option from the returned results by Google. Finally, the experiments showed the information that was obtained using the proposed method is twice more accurate than the information that was obtained at the stage of automatic construction of ontology from Wikipedia.

Keywords: Ontology; Improvement and Development of Ontology; Bootstrapping Method; Google Search Engine; Wikipedia.

1. Introduction

Nowadays, the Web is considered a live entity that is growing and evolving fast over time. The amount of content stored and shared on the web is increasing quickly and continuously. Problems and difficulties such as finding and properly managing all the existing amount of information, arise as a consequence of this extensive development. To overcome such limitations the only possible way is to promote the use of Semantic Web techniques (Cantador et al. 2007). Ontologies are the basis and foundation of the Semantic Web. Ontology is a conceptual model which formally and explicitly simulates actual entities and the relations among them in a particular domain (Gruber 1993; Staab and Studer 2004).

Ontologies have been useful in lots of applications such as knowledge management, information retrieval, and question answering systems. They are considered as the basis and foundation of many new intelligent systems. Manual ontology construction is very costly, tedious, and error-prone. They also suffer from rapid aging and low coverage. The manual construction of ontology needs a lot of experts in particular domain and many annotators must work together for a long time (ShamsFard and AbdollahZade 2002). A few ontologies have been built manually the most famous of which are WordNet (Fellbaum 1998), Cyc (Lenat 1995) and Gene Ontology (GOC[1] 2000). Consequently, in recent years, one of the main challenges for researchers has been the automatic construction of Ontology. One of the main problems of automatic ontology construction is the incompleteness of information required to construct that ontology; as the web corpora from which ontology is extracted, do not contain all information related to the given domain of ontology. In addition, during the automatically ontology construction process, certainly not all information can be fully extracted from web corpora.

The aim of this study is to provide a strategy for development of the ontologies. Therefore, this study first discusses the automatic construction of a prototype ontology in animal domain with the help of articles in Wikipedia. Hence, consequently an ontology is generated automatically with the use of semantic relations obtained in the structure of Wikipedia template pages, Infoboxes, and their hierarchical categories. Next, a Bootstrapping method is proposed to improve the constructed ontology and complete its information using the extracted information in ontology. Our method can automatically extract new information and extend the initial ontology with the help of Google search engine.

The rest of this paper is as follows: in section 2, the related work is described. In section 3 and 4, the technique

[1] Gene Ontology Consortium

* Corresponding Author

for the automatic construction of prototype in Persian ontology will be discussed and also the proposed solution to improve the ontology constructed with the use of Bootstrapping techniques will be explained. In addition, in these sections, experiments carried out to evaluate the proposed method will be described. In section 5, evaluation of proposed method has been presented and finally, section 6 draws conclusions and offers some solutions for future work.

2. Related Work

In this section, researches on automatic ontology construction, extraction concepts based on predefined patterns, ways of developing concepts of a collection based on Bootstrapping techniques, and semi-supervised solutions are briefly presented.

2.1 Automatic Ontology Construction

Kylin system (Wu and Weld 2007) is a self-supervised learning system whose main idea is automatic construction of ontology using Infoboxes in Wikipedia pages and then creating Infoboxes for all Wikipedia articles. Another purpose of Kylin is automatic production of links to Wikipedia articles.

KOG system (Wu and Weld 2008) is an autonomous system for creating a rich ontology from Wikipedia pages. It uses statistical-relational learning techniques for combining Wikipedia Infoboxes with WordNet. It also uses Markov Logic Network (MLN) (Richardson and Domingos 2006) and the proposed solution "joint-inference" to predict Subsumption relationships between Infobox classes; while simultaneously mapped the classes to WordNet nodes. As a result, the constructed ontology contains Subsumption relations and mappings between Wikipedia's Infobox classes to WordNet.

YAGO (Suchanek et al. 2008) is a high quality ontology with a high coverage that consists of 1 million entities and 5 million facts. YAGO system combines category labels and Infoboxes in Wikipedia pages with WordNet nodes and in this way, a wide ontology is created automatically by using heuristic methods and rule-based techniques.

Another automatic ontology extension method were proposed based on supervised learning and text clustering. This method uses the K-means clustering algorithm to separate the domain knowledge, and to guide the creation of training set for Naïve Bayes classifier (Song and et al 2014).

Sanabila et al. automatically built a wayang ontology from free text. The information or knowledge that is contained within the text is extracted by employing relation extraction. This method was extracted instance candidates that were subsequently clustered using relation clustering (Sanabila and Manurung, 2014).

In other paper, an automatic approach was proposed based on Ontology Learning and Natural Language Processing for automatic construction of expressive Ontologies, specifically in OWL DL with ALC (Horrocks et al., 2007) expressivity, from a natural language text. The viability of their approach is demonstrated through the generation of complex axioms descriptions from concepts defined by users and glossaries found at Wikipedia (Azevedo et al. 2014).

2.2 Extracting Concepts Based on Pattern

Marti A. Hearst used Lexico-Syntactic patterns to extract Hyponyms relationships in natural language (Heast 1992). In this method, first, some pre-defined patterns by humans were considered. Next, by matching these patterns, the concepts and relations among them were extracted.

In another approach, a category system with large scales was created from category labels in Wikipedia pages. In order to find the "is-a" relations among category labels, methods based on connectivity in the network and Lexico-Syntactic Matching (Ponzetto and Strube 2007) were used.

Rion Snow and his colleagues proposed a new algorithm for automatic learning of hyponym (is-a) relations from text (Snow et al. 2005). Their main goal was automatic detection of Lexico-Syntactic patterns. First, they extracted concepts in the text using a small collection of manually defined patterns with regular phrases. Then, using dependency path feature obtained from parse tree they presented a public and all-purpose formula for these patterns. The proposed algorithm can automatically extracts the useful dependency paths and use them for other texts as well as for detection of new hyponym pairs.

In Sprat (Maynard et al. 2009) and SOFIE (Suchanek et al. 2009), a collection of concepts and relations was extracted from Wikipedia texts using rule-based techniques and with the help of some pre-defined patterns.

In another study, a semi-automatic approach is presented to build an ontology for the domain of wind energy which is an important type of renewable energy with a growing share in electricity generation all over the world. Related Wikipedia articles are first processed in an automated manner to determine the basic concepts of the domain together with their properties. Next the concepts, properties, and relationships are organized to arrive at the ultimate ontology (Küçük and Arslan, 2014).

Xiong et al. (Xiong et al. 2014) presented a semi-automatic ontology building method to build marine organism ontology used the role theory to describe the relations among marine organisms. After the realization of the ontology concept and relation extraction using ontology learning technology, a manual review, screening and proofing, then the ontology editor by using Hozo is required (Kozaki et al. 2002).

2.3 Extending the Concepts of a Collection

DIPRE system is a bootstrapping system that uses (Author, Book) pairs to extract structured relations from a large collection of web documents about books and authors (Brin 1998). In this approach, using five initial data as seeds, requests are sent to Google search engine and then the results are examined. The patterns consist of author and book pairs. Using the detected patterns and sending new requests to Google search engine, more information is extracted. Finally, the process of search and finding patterns are repeated and the data set is extended in this way.

Snowball is another system that includes a new strategy for producing patterns and extracting multi entities from plain-text documents whose main idea is similar to DIRPE (Agichtein and Gravano 2000). Actually, by developing key factors of DIRPE solution, the quality of obtained patterns without intervention of humans is calculated by the Snowball system in each iteration of the extraction process and better patterns are used in the next iterations.

SRES has a more complex model than DIRPE and Snowball (Rozenfeld and Feldman 2008). SRES system, in addition to using simple patterns for extracting relations, can also use more general patterns which have been defined in KnowItAll (Etzioni et al. 2005).

In another research, a distant-supervision system was proposed that uses the large semantic web database Freebase (Boolacker et al. 2008) as the seed and extracts new entities (Mintz et al. 2009). Each sentence used as the seed consists of a pair of entities which have participated in a relation in Freebase.

Carlson and his colleagues proposed a semi-supervised learning method for extracting information (Carlson et al. 2010). The main purpose of this method to extract new instances from the concept category and the relations among them using an initial ontology.

Another semi-supervised bootstrapping categorization method were used for retrieving the images related to medical terms from web documents (Chen et al. 2012). This method starts with a positive image for each term as a seed and continues the search process in an iterative way. New images extracted are also used in the next search process as the seed.

Yao et al. converted web data into semantic web descriptions that uses key-value pairs in JSON objects (Crockford 2006). Meanwhile, it builds semantic models for data instances, which can be applied to further semantic reasoning applications. Their used this method to extract schemaless JSON data automatically, including concepts, properties, constrains and values, and build semantic ontology to describe the metadata and instances (Yao et al. 2014).

3. Proposed Method

In this paper, first a method is presented for the automatic construction of prototype ontology using the structures of Persian Wikipedia pages. Since the ontology may contain incomplete information, another method is presented for solving this problem which can generally be used for improving and extending all types of ontologies. Figure 1 shows an overview of the proposed method for the ontology construction. The part above the dotted line shows information extraction process from Wikipedia and the automatic construction of prototype ontology using the existing structures in Wikipedia pages. The part below the dotted line shows the process for improving the constructed ontology using Google search engine.

The extraction method from Wikipedia and the automatic construction of prototype ontology are explained in the following subsections in more detail:

3.1 Proposed Ontology Construction Method

In this study, in order to construct the prototype ontology, information in Infoboxes and Navboxes in Wikipedia pages is used to extract the triple of facts (Extracting concepts). The information in the Navbox is used to extract category hierarchy between the given entities (Extracting relations). The various parts of Wikipedia are displayed in Figure 2. Wikipedia is a Web encyclopedia whose updated information can be accessed by users in different languages. Wikipedia articles are graphical in which related pages are linked to each other.

3.1.1 First Step: Wikipedia Pages Crawler

Pages relating to the fauna from the Persian part of Wikipedia were collected by this unit according to the predefined domain. The crawler acts in a way that at the first step receives an address as the starting page, then through the links on the page, collects further pages. At this stage, approximately 3,200 pages have been identified and saved by the crawler. This collection also consisted of unrelated pages, too.

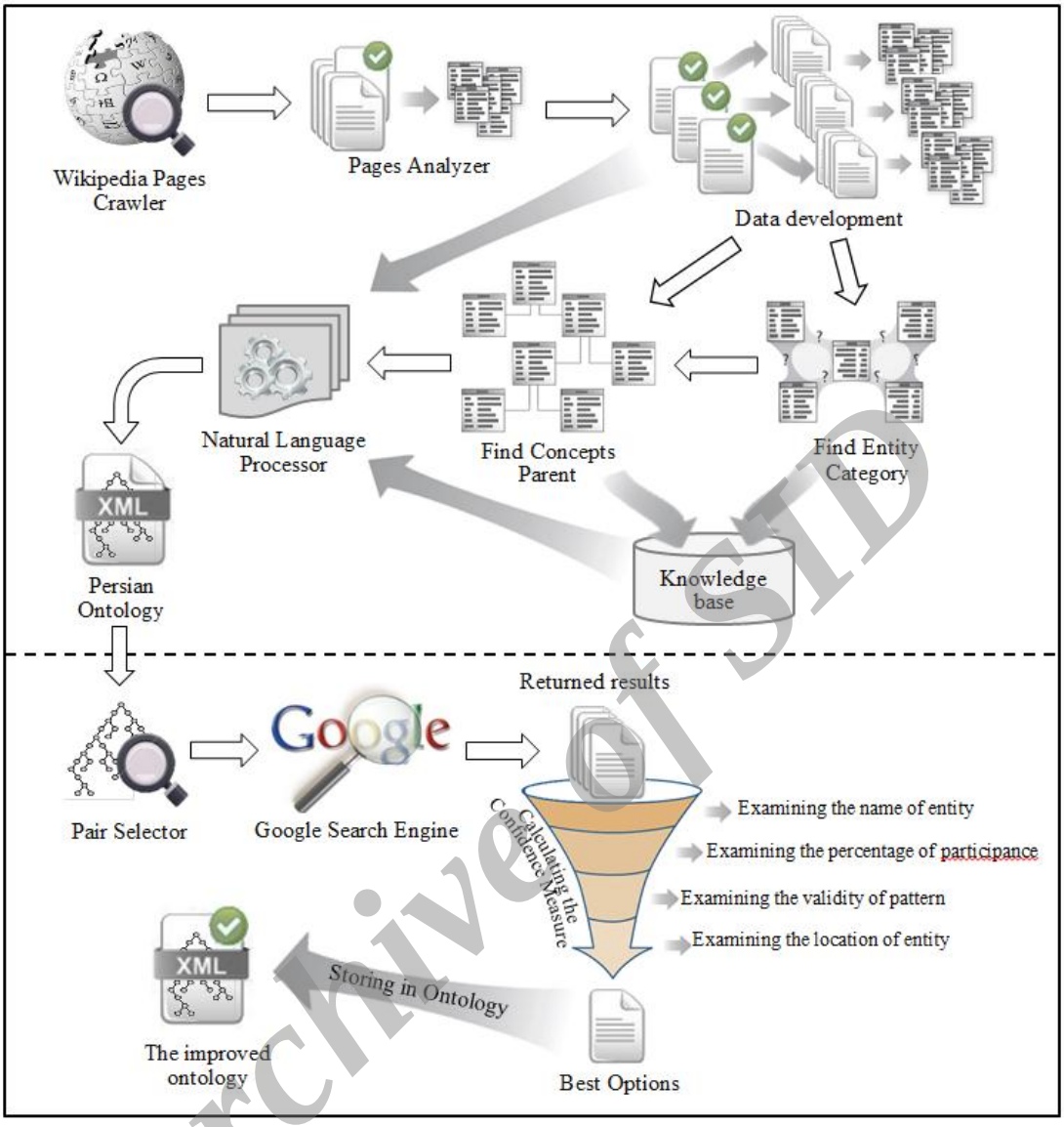


Fig. 1. The process of domain ontology construction using the existing structure in Wikipedia pages

(a) (b)

Fig. 2. (a) Navbox and (b) Infobox sample

3.1.2 Second Step: Wikipedia Pages Analyzer

This unit first examines the content of collected pages by Wikipedia crawler and then separates the template pages among them. Template pages are pages whose titles start with the word "olgô: / template:" and contain Navbox. To identify template pages relating to the fauna, page categories was also used. The page analyzer then extracts existing data in Navboxes. At this point, the proposed system from 3,200 extracted pages by the crawler have identified 11 template pages tailored to specify different conditions. Among the 11 existing Navboxes on these pages, 1,039 entities were also extracted.

Moreover, due to the existence of duplicate entities in different Navboxes and in order to achieve the best results and avoid ambiguity in the next steps, the entity tooltips were also considered in a way that if the entity include more clear, they can be used. For example in the Persian Wikipedia pages, there is the word "râh-râh / stripes" in both Navboxes relating to the sub-families of the "h æ vâsiliân / Ardeidae" [1] and "joqd-hâ / Owls". The tooltips related to these creatures show their full description, "h æ vâsil-e râh-râh / striped Heron" and "joqd-e bigôsh-e râh-râh / without ear striped Owl".

3.1.3 Third Step: Data Development

Each link in the Navbox refers to a Wikipedia page that has an article. These pages may also have Navboxes related to animals. Therefore, all these pages are also investigated and if new Navboxes exist, their contents will be extracted. There may be no pages available for some data. In this situation, those data will be marked for applying certain procedures in the next steps. Finally, after completing this step, 44 Navboxes with 2,346 unique entities were extracted from attributes collection.

3.1.4 Fourth Step: Find the Category of Entities

In this step, the extracted entity categories are found. Each entity in the classification hierarchy of animals has a category of its own; for example "Mohredârân yek zirshâxe az T æ nâbdârân æ st / Vertebrata [2] is a Subphylum[3] of Chordata[4]". This relation is actually the same as is-a relation in the classification of animals. First, In order to find the category of extracted entities, a series of relations were achieved using a simple statistics of the information in the Infoboxes and by choosing the category with the most frequency for the entity.

Next, to find the remaining entity categories (entities whose categories have not been yet found and entities that were marked in the previous step due to not having a relevant page) the location of word in Navbox will be used. To do this, the neighbors of an entity are examined by traversing Navbox and the category of entities will be guessed by using its neighbor's categories.

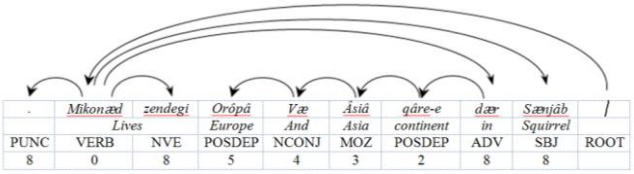
3.1.5 Fifth Step: Find Concept's Parent

In order to create a classification hierarchy of the extracted entities, finding the parent of each entity is necessary. This step is also performed in two phases: First, the concept parents are extracted through existing classification in Navboxes. Then to find the remaining concepts parent (concepts whose parents have not been found yet), the existing hierarchy in Infoboxes will be used to extract an integrated hierarchy. Finally, the information or the extracted metadata is stored in the Knowledge base (KB).

3.1.6 Sixth Step: Natural Language Processor

This unit, extracts the features associated with each entity using the existing texts in the Wikipedia pages and the metadata contained in KB. In order to extract features for each entity, five features including living location, food, size, weight, and longevity were considered. These features will be extracted using the rule-based approach (Maynard et al. 2009; Suchanek et al. 2009; Miháltz 2010). The defined patterns are given in Table 1.

At this point, after preparing the input file, the MateParser (Bohnet 2010) was applied. Below is an example to acquire the living location attribute of a Squirrel entity by using the dependency tree of sentence "Sænjâb dær qâre-e âsiâ væ orôpâ zendegi mikonæd / Squirrel lives in Asia and Europe continent".



In this case, the living location attribute of Squirrel is obtained through pattern 1 in Table 1 and the rules contained in the parse tree, is achieved as "qâre-e âsiâ væ orôpâ / Asia and Europe continent" noun phrase, given that the adverbial preposition "d æ r / in" comes to "zendegi mikonæd / Lives" verb.

3.1.7 Seventh Step: Ontology Producer

Finally, the obtained collection of entities and their relations were stored in an XML file. The resulting ontology includes a hierarchy of animal classification and five attributes related to each entity.

[1] Herons
[2] Vertebrates
[3] Sub-branch
[4] Chordates

Table 1. Defined patterns for extracted features

Pattern No.	Defined patterns
----------	---------- ***Living Location Pattern*** ----------
1	dær NP [zendegi mikonæd | pærâkænde æst | yâft mishævæd | sâken æst | sokônæt dâræd] [lives | scatters | finds | dwells | resides] in NP
2	[bômi | zistgâh | sâken] [dær | ""] NP æst is [native | habitat | residing] [in | to | ""] NP
3	mæhæle [sokônæt | zendegi] NP æst [life | residence] location is NP
4	dær NP [miziæd | ziste | mizist] [living|lived] in NP
----------	---------- ***Feed Pattern*** ----------
5	æz NP tæqzie mikonæd feeds of NP
6	[qæzây-e | xôrâk] ânhâ [shâmel | bishtær | ""] NP æst their [food | feed] is [included | more | ""] NP
7	rejim-e qæzâie [æz | shâmel] NP [tæshkil mishævæd | æst] diet [consists of | is] NP
8	[æz | ""] NP [râ | ""] mixoræd eats NP | NP to eat
----------	---------- ***Size (Length) Pattern*** ----------
9	tôl | derâzâ | qæd | qâmæt | bolændi | ændâze] NP [milimetr | sântimetr | metr | s.m | s | m] [æst | dâræd] [length | long | stature | height | size] [is | have] NP [millimeter | centimeter | meter | mm | cm | m]
10	NP [milimetr|sântimetr|metr|s.m|s|m] [tôl|derâzâ|qæd|qâmæt|bolændi|ændâze] dâræd have [length|long|stature|height|size] NP [millimeter|centimeter|meter|mm|cm|m]
11	tâ NP [milimetr | sântimetr | metr | s.m | s | m] roshd mikonæd grows up to NP [millimeter | centimeter | meter | mm | m | cm]
----------	---------- ***Weight Pattern*** ----------
12	[væzn | josse] NP [miligæræm | gæræm | kilogæræm | k.g | k | g | mg] [weight | body] is NP [milligram | gram | kilogram | mg | m | kg]
13	NP [miligæræm | gæræm | kilogæræm | k.g | k | g | mg] væzn dâræd weights NP [milligram | gram | kilogram | mg | m | kg]
----------	---------- ***Longevity Pattern*** ----------
14	[miângin | motevæset | ""] tôl omr NP [rôz | mâh | sâl] [average | mean] life time is NP [day | month | year]
15	NP [rôz | mâh | sâl] omr [dâræd | mikonæd] have life time NP [day | month | year]

3.2 Experiments of Ontology Construction Method

In order to evaluate the automatic construction of ontology, the experiments were performed on 3,200 Wikipedia articles saved by crawler in 30/1/2014.

To perform the experiments, 100 instances of Wikipedia articles were randomly selected and manually annotated. The evaluation of the automatic ontology construction system was done separately for calculating the accuracy of the extracted rank of the entity, accuracy of the extracted parent of the entity, accuracy of the extracted hierarchy and the accuracy of the extracted attributes for each entity. Table 2 shows the accuracy measure in the subsections evaluated. The results of the proposed method were compared with the structure of Carol Linnaeu's classification (Swedish botanist, physician and zoologist), introduced in his famous book "Systema Naturae" (Linnaeus 1735). It should be noted that this accuracy measure is one of the most popular measures in evaluation of algorithms. In some cases (concept parent extraction and hierarchy extraction), we have changed the classic accuracy measure in order to better evaluate our proposed method. The detail of this customized measure is described in more detail.

In order to evaluate the accuracy of the extracted rank of entities (row 1 of the table 2), if the rank of the entity has been extracted correctly, 1 and otherwise 0 will be considered as the score. In order to evaluate the accuracy of extracted parents of entities (row 2 of table 2), if the parent of the entity has been extracted correctly, 1 will considered as the score and otherwise for each generation distance with the parent, 0.2 will be subtracted from the score; this mean that if the proposed system, wrongly tags the grandfather of an entity as the parent of the entity instead of his father, its score will be 0.6. In addition, for evaluating the accuracy of the extracted hierarchy for each entity (row 3 of table 2), the location of each entity in the hierarchy of category of animals will be examined. If categorized correctly, 1 and otherwise 0 will be considered as the score.

Table 2. The results of experiments (accuracy criterion)

Subsection	Accuracy (%)
Data category extraction accuracy	93
Concepts parent extraction accuracy	89.8
Hierarchy extraction accuracy	99
Defined patterns accuracy to extract features	91

In the end, in order to evaluate the accuracy of the extraction of attributes, it is clear that with having 100 articles selected randomly and five attributes defined for each attribute (Location, Nutrition, Length, Weight and longevity), 500 attributes are evaluated. If an attribute is correctly extracted, its score will be 1 otherwise 0. Since most of the Persian Wikipedia articles in animal fields do not contain all five attributes, the constructed ontology in the section of attributes extractions has a very little information. According to experiments, from 500 attributes evaluated, only 67 attributes were in the pages and 91% of these were extracted correctly (Row 3 of Table 2); it means 61 attributes were extracted correctly and 6 were extracted incorrectly or they existed in the page or were not extracted by the pattern. In other words, only 13.4% of the attributes existed in the Wikipedia articles and of this, 12.2% of attributes were correctly extracted because of selecting the proper patterns. Even though, 12.2% is not an acceptable value for the attribute extraction subsection. The reason is the insufficient information in the Wikipedia articles, therefore 91% for the accuracy of the patterns defined in attribute extraction section is a considerable value which indicates the proper performance of the proposed algorithm in the attribute extraction section. As a result, in order to solve the problem of insufficient information in Wikipedia articles, it is necessary to improve the constructed ontology.

4. Improvement Method

As stated previously Wikipedia does not have complete information about all of the extracted data (Either there is no sentence related about a given attribute in the page or no pages have been defined for a given data). On the other hand, the proposed system will definitely not be able to extract whole attributes in the page during the ontology construction. Therefore, in order to improve the ontology and complete its information, a solution based on bootstrapping method were suggested so that new information can be extracted using the exiting information and with the help of Google search engine.

4.1 Proposed Ontology Improvement Method

To do this, whenever any of the attributes related to an entity, does not exist in the initial ontology (i.e., it was not extracted from Wikipedia pages in the step of automatic construction of ontology). In such cases, one or more commands are sent to Google search engine to extract the value of that attribute. In order to send the search words to Google, the (entity, attribute title) pair is used. Table 3 shows the words sent to Google search engine in case that any of the five attributes in the initial ontology is missing. For example, if the living location of an animal has not been extracted from Wikipedia pages in ontology construction stage, four commands in the pattern available in Table 3 will be sent to Google search engine. In the end, the first ten responses from Google will be evaluated and analyzed. The process of analysis and evaluation will be done on the ***snippet part*** of the returned results. Part of the return results from Google search engine is shown in Figure 3. The red rectangles in this figure are two samples of snippet part.

Since there might be different responses (relevant or irrelevant) in the results from Google search engine, a confidence measure is considered for selecting the best option and measuring the accuracy of the result which will be calculated from the sum of the following criteria and using the proposed algorithm discussed in Figure 4.

- *Name examining measure:* This measure has been in fact considered for pre-processing; it means that after receiving the results from Google, each snippet without the complete name of the data is removed so that it will not be processed in the next step.
- *Participation percentage measure:* This measure is considered for the percentage of participation of the entity in the returned results from Google. If the name of the data appears completely in more results, it can be said that the extracted result is correct with a higher confidence. Assuming that, TF_{Entity}, is frequency of the snippets in which the name of the entity exists and, $NR_{GoogleSearch}$, equivalent to DF measure, is the number of the returned results from Google (here the first ten results are evaluated), the participation percentage measure, $P_{Participation}$, is calculated as follows, which is similar to TF.IDF measure (Manning et. al. 2008):

$$P_{Participation} = \frac{TF_{Entity}}{NR_{GoogleSearch}} \times 100 \qquad (1)$$

- *Pattern accuracy measure:* A score is assigned to each pattern in Table 1 based on the percentage of their participation and the amount of correct results extracted in the automatic ontology construction stage; it means that the pattern is more accurate if it has higher participation and the number of its incorrect obtained results is smaller. Scores and ranks for the patterns in Table 1 are assigned in the following way: assuming that, $TF_{Pattern}$, is the frequency of a given pattern for extracting the attribute in the construction stage, $P_{Correct}$, is the percentage of correct results and, $P_{Incorrect}$, is the percentage of incorrect results extracted from the same patterns, the score of each pattern, P_{Score}, is calculated based on equation 2 (Han et. al. 2011).

$$P_{Score} = TF_{Pattern} \times (P_{Correct} - P_{Incorrect}) \qquad (2)$$

It is important to note that when comparing two patterns, the higher is the, P_{Score}, for a pattern, the more valid is the pattern. Therefore, the patterns in Table 1 are ranked based on this measure and one round of scores normalization (normal distribution) in this way, pattern measure, $M_{Pattern}$,

will be obtained. The reason for the ranking is that whenever the value of an attribute is extracted based on a pattern; the amount of accuracy of the pattern can be measured and used in calculating the confidence measure.

Table 3. The words sent to Google search engine for the five attributes in ontology

	Attribute title	Query words
1	Mækân zendegi mojôdiyæt Entity living location	1- Mojôdiyæt + "Zendegi Mikonæd" (Entity + "Lives") 2- "Zistgâh" + Mojôdiyæt ("Habitat" + Entity) 3- "Mækân zendegi" + Mojôdiyæt ("Living Location" + Entity) 4- Mojôdiyæt + "Bômi" (Entity + "Native")
2	Khôrâk mojôdiyæt Entity nutrition	1- "Khôrâk" + Mojôdiyæt ("Nutrition" + Entity) 2- "Ghæzâye" + Mojôdiyæt ("Food" + Entity) 3- Mojôdiyæt + "Mikhoræd" (Entity + "Eats")
3	Ændâze Mojôdiyæt Entity size	1- "Ændâze" + Mojôdiyæt ("Size" + Entity) 2- "Tôl" + Mojôdiyæt ("Length" + Entity)
4	Væzn Mojôdiyæt Entity weight	1- "Væzn" + Mojôdiyæt ("Weight" + Entity) 2- "Josse" + Mojôdiyæt ("Bulk" + Entity)
5	Tôl Omr Mojôdiyæt Entity longevity	1- "Tôl Omr" + Mojôdiyæt ("Longevity" + Entity)

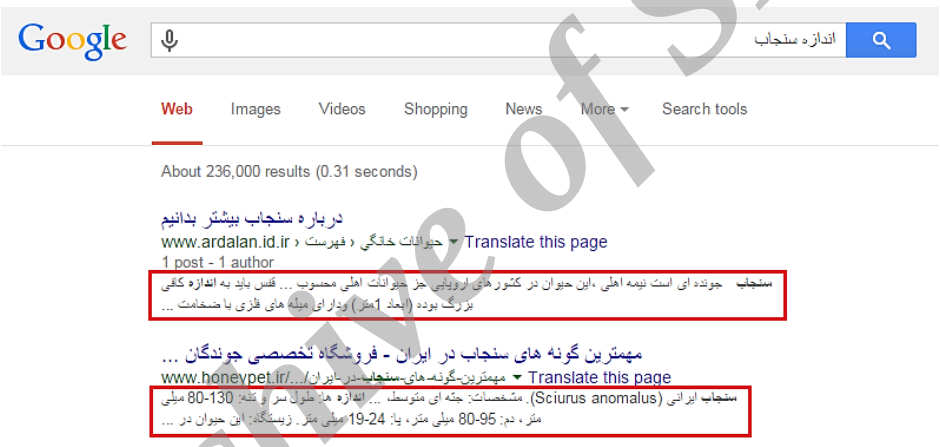


Fig 3. Part of the return results from Google search engine to query "Ændâze Sænjâb / Size of squirrel"

- *Location of presence measure:* If both pairs (the entity and the title of the attribute) appear in a sentence simultaneously and match the pattern, the confidence measure is considered 100%. The more is the distance, the less will be the confidence. Here, for each sentence distance between the pair, 20% is decreased from its score and this will be done for two sentences before and after the sentence matched with the patterns. This decay factor has been achieved by experimental results.

In the end, the best option will be obtained by the algorithm proposed in the Figure 2. Using the proposed algorithm, whenever there are multiple different options for the selection of an attribute in the results returned by Google search engine, the best option can be selected by the confidence measure and its accuracy can be measured, too. Here, based on our empirical evaluations, the results whose confidence measure is below 30% ($M_{Confidence} < 30\%$), will not be considered due to the lack of confidence in the accuracy of the result.

Table of pattern, TP, in algorithm 1 includes columns for defined pattern, $Pattern_i$, the score of the pattern, $Score_i$, the average of scores for similar patterns, Avg_{Score}, and the amount of standard deviation in proportion to the similar patterns, Var_{Score}. Similar patterns refer to the patterns defined for extracting an attribute. These values are calculated in advance and placed in the columns of the table. As stated above, the score of each pattern, $Score_i$, is calculated by equation 2 and the value of, Avg_{Score}, is calculated by calculating the average of scores of similar patterns. Similarly, the value of, Var_{Score}, is calculated via equation 3, which is the classic equation for computing variance (Han et. al. 2011):

$$Var_{Score} = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(Score_i - Avg_{Score})^2} \qquad (3)$$

The search command with two words (entity e and the title of the attribute a) is first sent by Algorithm 1 in Figure 4 to Google search engine. Then results returned from Google are examined. If any of the snippets does not have the name of the entity completely, further processing on the snippet will be ignored and the operation will go on for the next

snippet. In the next step, if each sentence in the snippet matches a pattern in the table of patterns, the Confidence Measure function will be called with the sent results. This will be done for all sentences from the extracted snippets. In the end any sentence with the highest confidence measure will be selected as the best option for the given attribute.

Algorithm 2 first finds out if both search words are present in a sentence. If this is the case, the confidence measure is 100%; because it can be definitely said that the returned result is completely true. If both of the search words are not present in a sentence, the average of participation percentage of entities in the returned result of, Google $P_{Presence}$, (Percentage of Participation function), the amount of accuracy of the pattern, $M_{Pattern}$, (Measure of pattern function) and the distance of search words, $L_{Presence}$, (Location of presence function) will be calculated as the confidence measure.

PercentOfParticipation function using relation 1, calculates the percentage of participation of the entity in the results returned by Google search engine. MeasureOfPattern function using equation 4, Normal standard distribution, calculates the accuracy of the pattern (Han et. al. 2011).

$$z = \frac{Score_i - Avg_{score}}{Var_{score}} \quad (4)$$

Then the value of z is obtained from the table of normal standard distribution and the percentage of the accuracy of the pattern is calculated.

LocationOfPresence function, as mentioned previously, considers the confidence measure of the sentence, 100% if both pairs (The entity and the title of the attribute) appear in a sentence at the same time. This value is calculated separately in the first line of the ConfidenceMeasure function (Algorithm 2 in Figure 4); because in this case, There is no need for calculating the other measures. Definitely the larger is the distance of the search pairs, the less will be the confidence percentage. Here, for each sentence distance between pairs, the scores will be decreased by 20%; it means that if the distance between a given pair is one sentence, the score will be 80% and if the distance is two sentences, the score will be 60% and so on. This will be done until two sentences before and after the sentence matched with the pattern and if the distance between pairs is more than two sentences, the score will be 30%.

4.2 Experiments of Ontology Improvement

The proposed method for improving the ontology focuses on extracting information using Bootstrapping methods for extending and developing the parts of the ontology which do not exist in the Wikipedia corpus; it means those 433 attributes out of the 500 attributes evaluated that were not extracted due to the incompleteness of texts in Wikipedia articles.

Algorithm1: AttributeExtractionUsingGoogleSearchEngine
Input: Entity *e*, Attribute *a*, Table of Patterns *TP*
Output: Attribute Value $Attrib_{val}$, Best Confidence Measure $BM_{Confidence}$

```
Results ← GoogleSearchEngine(e, a)
For each (Snippet in Results)
{
    If (Snippet not contains e) then Continue
    For each (Snippet.Sentence Matched to TP.Pattern_i)
    {
        Array[i].Confidence ←
            ConfidenceMeasure(e, a, TP.Score_i, TP.Avg_Score, TP.Var_Score)
        Array[i].AttributeValue ← Snippet.Sentence
    }
}
index ← Index of Maximum(Array.Confidence)
Return Attrib_val ← Array[index].AttributeValue and
       BM_Confidence ← Array[index].Confidence
```

Algorithm2: ConfidenceMeasure
Input: Entity *e*, Attribute *a*, Score of Pattern $Score_i$, Average of Pattern Scores Avg_{Score}, Variance of Pattern Scores Var_{Score}
Output: Confidence Measure $M_{Confidence}$

```
If ("e and a" Located in One Sentence) then Return M_Confidence ← 100%
Else
    {
        P_Participation ← PercentOfParticipation (e, a)
        M_Pattern ← MeasureOfPattern(e, a, Score_i, Avg_Score, Var_Score)
        L_Presence ← LocationOfPresence(e, a)
        Return M_Confidence ← ( ∑ P_Participation, M_Pattern, L_Presence)/3
    }
```

Fig 4. Algorithm for extracting information using Google search engine and selecting the best result (Algorithm1) and calculating the confidence measure (Algorithm2).

Finally, in order to calculate the percentage of improvement of the ontology, a new attribute was examined that was extracted by the proposed algorithm and did not exist in the initial ontology. Assuming that, $N_{Correct}$, is the sum of the number of new and correct relations extracted by the Google search engine and, N_{Total}, is the total number of relations that did not exist in these 100 random samples in the initial ontology, the

percentage ontology improvement, $P_{Improvement}$, is calculated by equation 5, which is standard accuracy measure for not covered samples in initial ontology:

$$P_{Improvement} = \frac{N_{Correct}}{N_{Total}} \quad (5)$$

5. Evaluation

Experiments were performed separately in two subsections for the automatic construction of prototype ontology and the improvement of the initial ontology. As a result, the percentage of the proposed method improvement can be evaluated by comparing the initial ontology and the improved one.

According to the experiments performed, out of the 433 attributes that did not exist in the initial ontology (due to the incompleteness of Wikipedia articles), 152 attributes were extracted by the proposed method for improving the ontology. 138 of these attributes were extracted, correctly. Table 4 shows the details of the calculation of the defined patterns accuracy.

The number indicates two interesting points; first, even with extending the domain of information collection in the web using Google search engine, the patterns defined in the section of extraction attributes from texts, still gain the 91% accuracy score. Second, the number of newly extracted attributes using the proposed algorithm is twice more than (2.26 times) the number of attributes that were extracted using the information available in texts of Wikipedia articles. This value can still be extended by increasing the number of retrieved documents from Google search engine. Due to the limitation in sending requests to Google search engine (a regular user can get only 10 results per order while only 100 requests are permitted per day), we were forced to evaluate only the top 10 returned results. By increasing this value, the domain of the extracted attributes can be extended considerably.

Another important issue to note is about the comparison of our method with other existing algorithm. Our method is the first study in ontology extraction for animal's domain in Persian, and due to this fact, we are unable to compare our work with similar researches.

Conclusions and Future Work

In this study, first a prototype ontology was automatically extracted from the existing structures in Wikipedia. Since the information in the constructed ontology was not complete (due the incompleteness of Wikipedia articles), a solution for improving and extending the information in the initial ontology was proposed using Google search engine and Bootstrapping method.

Considering the complexity of processing Persian language (due to the freedom in the order of words and the lack of strong rules), by selecting correct patterns for extracting attributes in the end, good results were achieved in the section of automatic construction of ontology. In fact, the resulting ontology is a domain ontology particular for animals which in addition to having a hierarchy of different animal categories, has attributes of living location, nutrition, size, weight and longevity for each of them. This ontology can be used for educational and training purposes. Also, considering the increasing growth of Web and the fact that many up-to-date information resources are being added to the Web, the method for improving ontology can be used to extend and update the constructed ontologies.

It is predicted that in the future, machine learning methods will be tried. Furthermore, a pattern bank will be defined so that in case of detecting new patterns in the process of extracting entities, they can be processed, examined and registered in the pattern bank. This is done so that the new patterns can be used in the process of extracting entities in the next steps. These steps are done iteratively so that each time the collection can be extended with new patterns.

Table 4. Details of the calculation of defined patterns accuracy to extract features

Methods	Total	Existing attributes	Correct	Incorrect
Automatic Construction of Ontology	500	67	61	6
Improvement of Ontology	433	152	138	14

References

[1] Azevedo, R. R., Freitas, F., Rocha, R. G. C., Menezes, J. A. A., Oliveira Rodrigues, C. M., and F. P. Silva, G. (2014). An Approach for Learning and Construction of Expressive Ontology from Text in Natural Language. Proceedings of 2014 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT), 149-156.

[2] Bohnet, B. Top accuracy and fast dependency parsing is not a contradiction. Proceedings of Coling 2010, 89–97.

[3] Crockford , D. JSON: The fat-free alternative to XML. Proceedings of XML 2006.

[4] Gruber, T. Ontolingua: a translation approach to providing portable ontology specs. Knowledge Acquisition, 5(2), (1993). 199-220.

[5] Han, J., Kamber, M., & Pei, J. Data mining: concepts and techniques: concepts and techniques. Elsevier. (2011).

[6] Horrocks, I. et al. OWL: a Description-Logic-Based Ontology Language for the Semantic Web. In: The Description Logic Handbook: Theory, Implementation and Applications, (2007). 458-486.

[7] Kozaki, K., Kitamura, Y., Ikeda, M. et al. Hozo: an environment for building/using ontologies based on a

fundamental consideration of 'Role' and 'Relationship'. Proceedings of Computer Science, (2002). 2473: 213-218.
[8] Küçük, D., and Arslan Y. Semi-automatic construction of a domain ontology for wind energy using Wikipedia articles. Proceedings of Renewable Energy 62, (2014). 484-489.
[9] Lenat, D. B., & Guha, R.V. Building Large Knowledge-Based Systems: Representation and Inference in the CYC Project. Boston: Addison-Wesley Publishing Co. (1990).
[10] Linnaeus, C. Systema naturæ per regna tria naturæ, secundum classes, ordines, genera, species, cum characteribus, differentiis, synonymis, locis. Sweden. (1735).
[11] Malo, P., Siitari, P., Ahlgren, O., Wallenius, J., & Korhonen, P. Semantic Content Filtering with Wikipedia and Ontologies. Proceedings of Third International Workshop on Semantic Aspects in Data Mining (SADM'10) in conjunction with the 2010 IEEE International Conference on Data Mining, (2010). 518–526.
[12] Manning, C. D., Raghavan, P., & Schütze, H. Scoring, term weighting and the vector space model. Introduction to IR, 100. (2008).
[13] Maynard, D., Funk, A., & Peters, W. SPRAT: a tool for automatic semantic pattern-based ontology population. Proceedings of the International Conference for Digital Libraries and the Semantic Web. Italy: Trento. (2009).
[14] Miháltz, M. Information Extraction from Wikipedia Using Pattern Learning. Journal of Acta Cybern. 19(4), (2010). 677-694.
[15] Ponzetto, S. P., & Strube, M. Deriving a Large Scale Taxonomy from Wikipedia. Proceedings of Association for the Advancement of Artificial Intelligence (AAAI07). (2007).
[16] Richardson, M., & Domingos, P. Markov Logic Networks. Machine Learning. (2006).
[17] Sanabila, h., and Manurung, R. Towards Automatic Wayang Ontology Construction using Relation Extraction from Free Text. Proceedings of the 8th Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities (LaTeCH), (2014). 128-136.
[18] ShamsFard, M. Persian text processing: past achievements, challenges ahead. Proceedings of the second workshop on research in English and computers, University of Tehran, (2006). 172-189.
[19] ShamsFard, M., & Barforoush, A. A. Extracting conceptual knowledge from text using linguistic and semantic patterns. Journal of Cognitive Science, 4(1), (2002). 48-66.
[20] Shibaki, Y., Nagata M., & Yamamoto, K. Constructing Large-Scale Person Ontology from Wikipedia. Proceedings of the 2nd Workshop on Collaboratively Constructed Semantic Resources, Beijing. (2010).
[21] Song, Q., Liu, J., & Wang, X. A Novel Automatic Ontology Construction Method Based on Web Data. Proceedings of Tenth International Conference on Intelligent Information Hiding and Multimedia Signal Processing, (2014). 762-765.
[22] Staab, S., & Studer, R. Handbook on Ontologies. (pp. 1-17). Springer: International Handbooks on Information Systems. (2004).
[23] Suchanek, F. M., Kasneci, G., & Weikum, G. YAGO: a large ontology from Wikipedia and WordNet. Journal of Web Semantics: Sci. Serv. Agents World Wide Web6, (2008). 203–217.
[24] Suchanek, F. M., Sozio, M., & Weikum, G. SOFIE: a self-organizing framework for information extraction. Proceedings of the 18th International conference on World Wide Web, New York, NY, USA. ACM, (2009). 631-640.
[25] Syed, Z., Finin, T., & Joshi, A. Wikitology: Using Wikipedia as ontology. Proceeding of the second international conference on weblogs and Social Media. (2008).
[26] Wu, F., & Weld, D. Autonomously Semantifying Wikipedia. Proceedings of CIKM07. Portugal: Lisbon. (2007).
[27] Wu, F., & Weld, D. Automatically Refining the Wikipedia Infobox ontology. Proceedings of WWW08. (2008).
[28] Xiong, J., Liu, Y., Wang, J., and Lan, Y., Research of Marine Organism Ontology Semi-Automatic Construction. Proceedings of the Open Cybernetics & Systemics Journal, (2014). 984-989.
[29] Yao, Y., Liu, H, Yi, J., Chen H., & Zhao X. An Automatic Semantic Extraction Method for Web Data Interchange. Proceeding of 6th International Conference on CSIT, (2014). 148-152.
[30] Yu, C., Cuadrado, J., Ceglowski M., & Payne J. S. Patterns in Unstructured Data: Discovery, Aggregation, and Visualization. Proceeding of NITLE. (2002).
[31] Zhou, G., & Zhang, M. Extracting relation information from text documents by exploring various types of knowledge. Proceedings of Information Processing and Management, (2007). 43: 969-982.

Sedigheh Khalatbari received the B.Sc. and M.Sc. degree in Software Engineering from University of Guilan, Rasht, Iran in 2012 and 2015 respectively. Her main research interests include Natural Language Processing, Text Mining and Semantic Analysis.

Seyed Abolghasem Mirroshandel received his B.Sc. degree from University of Tehran in 2005 and the M.Sc. and Ph.D. degree from Sharif University of Technology, Tehran, Iran in 2007 and 2012 respectively. Since 2012, he has been with Faculty of Engineering at University of Guilan in Rasht, Iran, where he is an Assistant Professor of Computer Engineering. Dr. Mirroshandel has published more than 30 technical papers in peer-reviewed journals and conference proceedings. His current research interests focus on Data Mining, Machine Learning, and Natural Language Processing.