OntoPopulis, a System for Learning Semantic Classes

Hristo Tanev
Joint Research Centre, European Commission
via Enrico Fermi 2749
Ispra, Italy
hristo.tanev@ec.europa.eu

Abstract

Ontopopulis is a multilingual weakly supervised terminology learning algorithm which takes on its input a set of seed terms for a semantic category and an unannotated text corpus. The algorithm learns additional terms, which belong to this category. For example, for the category "environmental disasters" the input seed set in English is *environmental disaster*, *water pollution*, *climate change*. Among the highest ranked new terms which the system learns for this semantic class are *deforestation*, *global warming* and so on.

Keywords: semantic classes, ontology learning, terminology extraction

1 Introduction

Ontologies are knowledge-representation models describing concepts of a domain, their properties and the semantic relations between them. These models are used in Natural Language Processing and other AI systems. Recently ontologies have been built and exploited predominantly in the area of biology and medicine. There are also many other domains for which they have been used: remote sensing, education, environment and security, etc.

The most fundamental building block of the ontology model are the concepts of the domain. Each concept has lexical representation, which shows how the concept is being referred at the level of a specific language. For example, the concept [TV-SET] in English can be referred to as: *TV*, *TV set*, *television receiver*, and *telly*. Words which describe ontology concepts form the lexical layer of the ontology.

Ontologies are created mainly manually by domain experts, however in populating their lexical layer, terminology learning algorithms have successfully been exploited. In this paper we will describe such a multilingual algorithm which given a set of ontology concepts learns new set of related concepts.

For each concept and language under consideration, the language experts define a small seed set of terms, belonging to the concept and its sub-concepts. For example for the concept *disaster* the seed set for English can be: *disaster*, *flood*, *earthquake*, *forestfire*, *wildfire*.

The seed set is then expanded by the algorithm by learning new terms, referring to the same main concept (*disaster*) and its sub-concepts, for example it will learn words like *calamity*, *tsunami*, *landslide*. These terms belong to the category *disaster* and its sub-categories *tsunami* and *landslide*

The algorithm was named *OntoPopulis* (ONTOlogy learning and POPULation).

In this paper we describe the algorithm and its application in the domain of environment for English.

2 Related work

The first algorithms for ontology learning from text started to appear about 20 years ago. Currently, many approaches are described in the literature; they are designed to learn generic and domain-specific ontology resources.

One of the first comprehensive overview of these approaches is presented by (Cimiano, 2006).

More recent surveys of ontology learning from text are presented in (Lourdusamy and Abraham, 2019) and (Al-Aswadi et al., 2020).

The approach in this paper is inspired by an earlier work, described in (Tanev and Magnini, 2006).

3 OntoPopulis

OntoPopulis is a multilingual algorithm for learning semantic classes. It does not use any language-specific tools or annotations. The algorithm accepts as an input a set of seed terms for each ontology concept under consideration and an unannotated corpus. For example, for the concept *disaster* the seed set is *disaster*, *wildfire*, *earthquake* and for the concept *environmental disaster* it is *environmental disaster*, *water pollution*, *climate change*

The algorithm performs two processing steps to learn the new lexical items for the input concepts: (i) feature extraction and (ii) lexical learning.

3.1 Feature Extraction and Weighting

For each category (e.g. environmental disaster), we consider left and right context features.

Each left context feature consists of uni-gram or bi-gram and can be followed by a preposition or another stop word. For example *primary cause of* is a left context feature; it occurs on the left side of words from the category *environmental disaster*. i.e. *primary cause of water pollution*

Similarly, right context feature appear on the right side of the seed terms, e.g. *and overfishing* is a right side context feature and it appears in phrases like *water pollution and overfishing*.

The left and right context features are weighted using a formula which considers the frequency of their co-occurrence with the seed terms.

Each context feature has to appear at least 3 times in the corpus with the seed set terms

For each such a context feature n and a semantic category C we calculate the score:

$$score(n,C) = \sum_{st \in C} PMI(n, st)$$

where $seeds(C)$ are the seeds terms of the category C and $PMI(n, st)$ is the point-wise mutual information which shows the co-occurrence between the feature n and the seed terms.

At the end of this learning phase there is a possibility for a linguist to perform manual feature selection from the list of the top ranked features. Manual cleaning is optional when high precision is the goal. In the reported experiments, however, we haven't used it.

Table 1 lists the top-ranked context feature for the semantic category environmental disaster

As one can observe, the context feature of OntoPopulis are very easy to evaluate semantically and linguistically. The table shows that these features come from semantic properties, predicates and related concepts of the considered semantic category.

3.2 Term Extraction

The term extraction and learning stage takes the features, which were learned and manually selected for each category in the previous stage and extracts as candidate terms uni-grams and bi-grams, which frequently co-occur with these features and which do not contain stop words, numbers or capitalized letters. Weighting of the candidate terms was carried out with the view to optimize the efficiency of the calculations. For this reason, we avoid to obtain the frequency of each candidate term in the corpus and we rather calculate the term feature vector in a non-standard way. It would be statistically more correct to use as a feature weight the point-wise mutual information between the term and the feature. However, this would require to collect statistics about the term frequency, which will decrease the algorithm speed. We weighted the term candidates, using the following algorithm:

1. For each category C we define a feature space, whose dimensions are only the features selected for this category

2. For each category C we define a category feature vector

   $$\vec{C} = (wf_1, wf_2, wf_3, ..., wf_{nc})$$

   where wf_i are the weights of the category features, calculated as $wf_i = score(n_i, C)$, where n_i is the n-gram used as i_{th} context feature in our model; $score(n_i, C)$ is calculated with the point-wise-mutual-information based formula presented in the previous subsection.

Feature	Score
threatened by X	1.38
X and land degradation	0.87
impacts of X	0.79
pollution and X	0.75
impact of X	0.59
emissions and X	0.54
X and greenhouse	0.54
emissions and X	0.54
X and greenhouse	0.54
land use and X	0.52
contributor to X	0.51
X and global warming	0.48
primary cause of X	0.45
combating X	0.44
worst effects of X	0.42
degradation and X	0.38
X and other environmental	0.37
contributes to X	0.32
exacerbated by X	0.32
reducing X	0.31
global warming and X	0.29
X and overfishing	0.29
exposure to X	0.28
warming and X	0.27

Table 1: Top-ranked features for semantic category environmental disaster

3. We normalize each category feature vector $\vec{C}$ by dividing its coordinates with its length and obtain its normalized form $norm(\vec{C})$ (this is needed when several categories are considered at a time)

4. Then, for each candidate term t for the category C we define a term feature vector $\vec{t_C} = (w_1, w_2, ..., w_{nc})$ where

$$w_i = \frac{f_i}{f_i + 3}$$

, where f_i is the frequency with which term t appears with context feature i.

5. The weight for each candidate term t for a category C is defined as a scalar product in the vector space defined for the category C, multiplied by the square root of the number of the non-zero features of the term feature vector:

$$weigth(t, C) = \vec{t_C}.norm(\vec{C}).\sqrt{NNZF(\vec{t_C})}$$

,where $NNZF$ returns the number of the non zero vector coordinates.

In plain words, this formula measures term suitability for a category by considering the co-occurrence of the term with the context features of this category and their weights.

6. Finally, the system orders the term candidates for each category by decreasing weight and filters out terms with a weight under a certain threshold.

4 Experiments

We run the Ontopopulis algorithm on a seed set of three words, modelling the concept *environmental disaster*: *environmental disaster*, *climate change*, *water pollution* The list of the highest ranked 27 newly learned terms is presented in table 2. The irrelevant ones are marked with asterisk. The relevant ones (77%)

shown in the table can be divided into two categories:

- hyponyms of the environmental disaster: global warming, deforestation, air pollution, acidification, rising sea, heat wave, ocean acidification, environmental degradation, desertification, warming temperatures, extreme weather, erosion, oil spills, habitat loss.
- factors, which cause environmental disasters: overfishing, illegal fishing, carbon emissions, greenhouse gases, noise.

The algorithm can be used in the process of building ontologies and semantic dictionaries for information extraction tasks, such as event detection, named entity recognition, sentiment analysis, etc. The algorithm can significantly speed up creation of language resources and it learns words which are rare and difficult to come up with by linguists.

In this paper we have evaluated this algorithm for English language, but it has no restriction on the language used, since it does not use any annotation or language-specific resources.

References

Fatima N Al-Aswadi, Huah Yong Chan, and Keng Hoon Gan. 2020. Automatic ontology construction from text: a review from shallow to deep learning trend. Artificial Intelligence Review, 53(6):3901–3928.

Philipp Cimiano. 2006. Ontology learning and population from text: algorithms, evaluation and applications, volume 27. Springer Science & Business Media.

Ravi Lourdusamy and Stanislaus Abraham. 2019. A survey on methods of ontology learning from text. In International Conference on Information, Communication and Computing Technology, pages 113–123. Springer.

Hristo Tanev and Bernardo Magnini. 2006. Weakly supervised approaches for ontology population. In 11th Conference of the European Chapter of the Association for Computational Linguistics, pages 17–24, Trento, Italy. Association for Computational Linguistics.

Term	Score
global warming	42.2
deforestation	21.4
overfishing	9.0
rising sea	8.8
∗*naturaldisasters*	8.1
∗*brexit*	5.4
∗*covid*	4.4
greenhouse gases	4.2
air pollution	4.1
acidification	3.9
heat wave	3.8
environmental degradation	3.6
rising temperatures	3.4
ocean acidification	2.9
environmental damage	2.8
∗*circumstances*	2.7
desertification	2.3
carbon emissions	2.2
illegal fishing	2.1
warming temperatures	2.1
extreme weather	2.1
erosion	2.1
∗*globalization*	2.0
noise	2.0
∗*fakenews*	2.0
oil spills	2.0
habitat loss	2.0

Table 2: Highest scored learned terms for semantic category environmental disaster