DEVELOPMENT OF EARTHQUAKE ONTOLOGY FOR PREVENTION AND PREDICTION

David Ramamonjisoa

Faculty of Software and Information Science, IPU, 152-52 Sugo Takizawa Iwate, Japan

ABSTRACT

This paper describes the method and development of earthquake ontology for prevention and prediction. In our previous work, we develop ontologies for organization knowledge such as university or faculty. We extend that work to construct the earthquake ontology. The aim is to build the ontology autonomously from freely available databases or textbooks. A preliminary result of the experiments is presented. A comparison of the human made ontology is discussed.

1. INTRODUCTION

Ontologies are engineering artifacts that define the formal semantics of the terms used, and the relations between these terms. They provide an "explicit specification of a conceptualization." Ontologies are used in order to specify the knowledge which are exchanged and shared between different systems including humans.

Manually building ontology is time and effort consuming and cannot be considered to build all the domains of the world. Ontology design and implementation from corpus or documents is investigated in various research disciplines such as text mining, knowledge acquisition from texts, and natural language processing. Results from these researches so far show that only taxonomical relationships ontology can be constructed autonomously with an accuracy at maximum of 40% as in Text2Onto [1] and OntoLearn [2]. OntoLearn has extracted the concept definition (gloss) automatically to the learned concepts. Some results from building ontology based on Wikipedia infobox and category tree can extract synonym, `class-instance` relationship, `is-a` relationship, and property information [3].

2. APPROACH

In this paper, we present the earthquake ontology construction. We applied some autonomously building framework to create the domain ontology. The approach is to automatically extract terms

from a set of representative documents and automatically structure the vocabulary. The approach is based on n-gram techniques to extract terms. Candidate terms are selected according to a given weight. Frequent terms are good candidates for domain ontology because they are representative of the domain. Concept hierarchy is built from the n-grams length in terms. The longest terms are considered as leaves of the hierarchy and their next upper level (hypernyms) built from the n-grams components or other terms. DBpedia ontology or category tree is used to set up the upper level hierarchy.

We used the the C-Value algorithm proposed in [4] to extract the concepts because it is the best one according to the evaluation result [5]. The measure of the C-Value is given as

$$C_{Value}(a)\begin{cases}\log_2|a|\cdot(f(a)) & if\ a\ is\ not\ nested\\ \log_2|a|\cdot\left(f(a)-\frac{1}{P(T_a)}\sum_{b\in T_a}f(b)\right) & otherwise\end{cases}$$

Where

a is the candidate string with |a|>1. C-Value does not work with a single word as concept.

f(a) is its frequency of occurrence in the corpus; |a| is the length of the string a, example: a=' w1 w2' |a|=2; Ta is the set of extracted candidate terms that contain a ; P(Ta) is the number of these candidate terms. C-Value is based on the frequency of a. The negative effect on the candidate string a being a substring of other longer candidate terms is reflected by the negative sign in front of the frequency of its substring of longer candidate terms Ta. The independence of a from these longer candidate terms is given by P(Ta). The greater this number is the bigger its independence.

Nested terms such as 'dip slip fault', 'strike slip fault', and so on are concepts and the 'fault' subconcepts. By using linguistic part-of-speech we can filter those candidates with the pattern (Adj|Noun)+|((Adj|Noun)*) Noun.

Other nested terms such as 'San Andrea fault', 'Alpine fault', 'Motagua fault', and so on are instances of the concept 'fault' and other subconcept of 'fault'. We can use the pattern ((Common Noun)*) Noun to filter them.

In the other side, we follow the classical methodology according to an expert in the domain and build the ontology from top level concept and detail each concept to attain the leaf concept of the domain.

3. ALGORITHM OF CONCEPT HIERARCHY BUILDING

Step 1: wrapping the source file

The source file is an article or text from the web. We are interested in the domain of earthquake so we collected documents in this domain freely available including Wikipedia sources. Html data are converted and processed to extract the texts and also important features such as term candidates

within html tags. PDF files are also converted into texts.

Step 2: tagging the texts with a natural language parser.

Step 3: extracting the strings that satisfy the defined patterns

Step 4: evaluating the C-Value of the string candidates

Step 5: arranging manually the result to have an ontology satisfying the user

Step 6: comparing the higher level concepts in the ontology with wordnet concepts [9]

4. EXTRACTION EXAMPLE

We use the Wikipedia page on 'fault_(geology)' : en.wikipedia.org/wiki/Fault_(geology) to show how the extraction algorithm works.

As of 2012/1/23, there are 166 matched with the word 'fault', however 18 terms in the source file are matched with the patterns composed with the term 'fault' at the end. After removing the redundant terms and normalizing each term, we obtained the following list:

{'strike-slip fault', 'detachment fault', 'listric fault', 'dip-slip fault', 'normal fault', 'alpine fault', 'san andreas fault', 'extensional fault', 'oblique-slip fault', 'seismic fault', 'wasatch fault', 'active fault', 'ring fault', 'thrust fault', 'transform fault'}

We applied a parser to get the POS for each word in the term. C-Value measure for each term is computed. The list below is obtained from the content of the article as type of faults.

{'dip-slip fault','strike-slip fault','oblique-slip fault','listric fault','ring fault'} are subconcepts of the concept 'fault', note that they are formed with the pattern (Adj|Noun + Noun).

{'detachment fault','normal fault','extensional fault', 'thrust fault','transform fault'} are subconcepts of the concept 'dip-slip fault'. c='left lateral strike-slip fault' or 'right lateral strike-slip fault' is a subconcept of 'strike-slip fault' and they are leaves of the concept tree. |c|=4 and C-Value(c)=2(2 – 1/1*1)=2. The greater the C-Value measure, the higher the position of the candidate in the concept hierarchy.

{'alpine fault','san andreas fault','wasatch fault'} are instances of the concept 'fault' because they are formed with the pattern (Proper Noun + Noun).

{'seismic fault','active fault'} are subconcepts of the concept 'fault' which are used to classify the faults according to their activity.

From this example, we can write axioms in description logic to form the ontology as below:

$$\left.\begin{array}{l}\text{'left lateral strike slip fault'}\\ \text{'right lateral strike slip fault'}\end{array}\right\} \subset \left.\begin{array}{c}\text{'listric fault'}\\ \text{'ring fault'}\\ \text{'strike slip fault'}\\ \text{'oblique slip fault'}\\ \text{'dip slip fault'} \equiv \text{'thrust fault'}\end{array}\right\} \subset \{\text{'alpine fault','san andreas fault','wasatch fault'}\} \in \textbf{'fault'}$$

5. DISCUSSIONS

Building ontology on a domain is time consuming and helping users to realize the model at a least

interaction is the final goal. Automating the burden, low level natural language processing and statistical operation are a must-have library for an ontologist.

This paper described a method to extract concepts and classify them hierarchically according to the page structure such as in html document and term extraction algorithm. Typical terminological structures such as adjective-nouns and proper noun- noun are used to classify a term whether a subconcept or an instance. General terms are distinguished from specific terms according to their length (number of subterms within the term).

Active faults are the most used to predict the earthquakes or volcanoes. When they are located in the subduction zone, they cause the 8+ magnitude ones when moving abruptly. Terms such as 'trench', 'tectonic plate', 'slab' and 'obduction' are used often to the earthquake ontology. Time scale in geology and radioactive emission such as radon in the atmosphere and other factors unrelated to the earth internal system are also important concepts that need to be studied further in order to have a good predictor [6][7][8]. An experiment within a large corpus is necessary to validate the proposed method and cover a domain such as earthquake and prediction ontology.

REFERENCES

Books

[1] Philipp Cimiano and Johanna Volker: *"Text2Onto: A Framework for Ontology Learning and Data-driven Change Discovery."* In NLP and Information Systems, LNCS Volume 3513, pp.257--271, 2005.

[2] P. Velardi et al.: *"Evaluation of OntoLearn, a methodology for automatic population of domain ontologies."* In Ontology Learning from Text: Methods,Applications, and Evaluation.IOS Press, 2005.

[8] S. Pulinets et al.:"*Ionospheric Precursors of Earthquakes*" Springer Ed, 2004

Journals

[6] Richard Kerr: *"After the Quake, in Search of the Science--or Even a Good Prediction"* Science Magazine, 7 April 2009, ol. 324 no. 5925 p. 322.

[7] S. Pulinets et al.:"*Lithosphere–Atmosphere–Ionosphere Coupling (LAIC) model – An unified concept for earthquake precursors validation"* Journal of Asian Earth Sciences 41, p.371–382, 2011.

Conference papers or contributed volumes

[3] Takahira Yamaguchi and Takeshi Morita: *"Building up a Large Ontology from Wikipedia Japan with Infobox and category Tree."* In Proc. Of the 3rd Interdisciplinary Ontology Meeting, pp.121--134, Tokyo, 2010.

[4] Katerina Frantzi et al. : *"Automatic Recognition of Multi-Word Terms: the C-value/NC-value Method."* In Proc. Of the 2nd European Conference ECDL, pp.585--604, Heidelberg, 1998.

[5] Scott Piao et al. : *"Evaluating Tools for Automatic Concept Extraction: a Case Study from the Musicology Domain."* In Proc. Of the Digital Futures Conference, Nottingham, UK, 2010.

Other resources

[9] Wordnet reference: http://wordnet.princeton.edu/