ATLANTIS PRESS

Advances in Computer Science Research, volume 58

Modeling, Simulation and Optimization Technologies and Applications (MSOTA 2016)

Text Classification Method Based on Machine Learning and Domain Knowledge Ontology

Zhiyong Gao[1], Shuhan Qiao[2,*] and Yongquan Liang[1]

[1]College of Information Science and Engineering Shandong University of Science and Technology, Qingdao, P. R. China

[2]School of Forest, Shandong Agricultural University, Taian, P.R. China

[*]Corresponding author

***Abstract*—The use of machine learning method is discussed herein to produce a corpus by domain knowledge ontology and conduct text classification according to the ontology of professional knowledge domain. Nowadays, a large number of literature materials have been accumulated in each professional field, and it is still in rapid growth. This constitutes a great challenge for researchers in various fields. To be specific, not only the workload in literature retrieval and reading is constantly increased, but also the work efficiency of the study is affected. In this paper, ontology is taken as the text feature extractor for storage, processing, classification and retrieval through ontology development tools Protégé, Jena and natural language processing tool NLTK, so as to facilitate the researcher for literature retrieval and reading. The advantage of this text classification method lies in that category structure is no longer a single tree structure, but instead, different categories may intersect and new category may be grouped by themselves.**

Keywords-machine learning; ontology; text classification

I. INTRODUCTION

In modern times, due to the rapid development of big data and Internet technology, the Internet information presents exponential growth, in which legal and administrative documents, scientific literature, etc. have accumulated to a very large number. Thus, it is impossible to read through the literatures in any field. Researchers in various fields have spent a lot of time retrieving and reading in the literature. Owing to insufficient understanding of the literature, some studies have become the work of "re-inventing the wheel", or unfamiliarity with legal and administrative document as well as case file causes fault in work. However, if the semantics of these literature data is machine-readable, we can get more information that are more accurate and more valuable from the data. Internet information sharing technology develops from the early plain text, HTML (Hypertext Markup Language) to XML (extensible markup language), from only text-available to be able to provide pictures, audio, video and other multimedia data, and from unstructured data to semi-structured data. With rich and multi-dimensional information, it has brought revolutionary change to people's work, study, life and the whole society. But most of these technologies serve people's reading and understanding, which is not conducive to the reading and analysis of computer programs. In order to enable direct exchange, sharing and retrieval of data between different computer systems, W3C (World Wide Web Consortium) firstly propose to use XML as the markup language for data exchange. XML, based on meta-data, could be customizable tags. XML provides a data type standard for data exchange between computer systems. Aiming at these problems, Tim Berners-Lee, in the XML2000 conference held in 2000, formally proposed the concept of Semantic Web and published "The Semantic Web" on the journal Scientific America in May 2001.

In this paper, through an example in the field of tax knowledge, the discussion is directed to how to construct a knowledge domain ontology and use the natural language processing software toolkit NLTK to process the text, including tax legal text, tax policy documents, tax inspection cases, and tax assessment cases. In addition, the machine learning function in NLTK3 is utilized to map the text into one of the ontologies (individual), use Jena to store in TDB and apply Query DL to inquire relevant information. The ontology construction tool is Protégé 5.1 and software development tool is Jena 3.1.

II. ONTOLOGY CONSTRUCTION

Ontology aims to acquire, describe and represent the knowledge in related fields, provide a common understanding of knowledge in the field, identify the commonly recognized terms in the field, and give explicit definition of the relationship between these words (terms) in the formal mode of different levels. Now, numerous researches are focused on how to establish the domain ontology method, like DEF5 method, TOVE method, TUM method, Stanford seven-step method and cycling method. However, a set of mature domain ontology construction method has not yet been shaped. We need to establish the construction method of domain ontology based on the composition and characteristics of the domain knowledge. In general, the domain ontology construction should have such several steps as determination of domain scope, collection and processing of domain knowledge resources, conceptualization, ontology construction, ontology evaluation, practical application, and feedback maintenance.

A. Ontology Language

W3C developed RDF/RDFS, OWL and other standards as ontology representation language of Semantic Web [2]. The basic concepts of RDF, a standard recommended by W3C to describe resources, are resources, attributes and statements. Such resource can be anything we can think of, like a text, a number, an entity, and a concept. Resources may be just a concept, which is not necessarily to be real. Attribute is a

special kind of resource for describing the relationship between resources. RDF takes XML as its syntax, which is conducive for the computer to automatically read or generate RDF due to the widespread use of XML on the Internet. But, RDF and XML are not the same, in which RDF refers to a data model. RDF can provide the explicit expression of the relationship between entities, which is unavailable for XML documents.

Although RDFS extends RDF in semantic expression, it still has a lot of semantic limitations, mainly covering: non-intersection of definition class, Boolean combination of classes, cardinality constraint, and special properties of attributes (transitivity, uniqueness, inverse attribute). Thus, W3C has developed a network ontology language OWL to further expand the support for concept semantics. Ontology language is used for the formal description for the display of domain model. The main needs of ontology language include well-defined grammar, efficient reasoning support, formal semantics, sufficient expression ability and convenience of expression. Under the current technical conditions, it is unlikely to meet all the needs of ontology language. In order to meet the different needs of specific application, OWL defines three sub-languages: OWL Full, OWL DL and OWL Lite.

1) OWL full

OWL Full uses all of the primitives that allow any combination of these primitives with the RDF/RDFS language. The OWL Full documents are legal relative to any legal RDF documents - RDF fully upward compatible. All valid RDF/RDFS inferences are valid Full OWL inferences. The disadvantage is that the expression of OWL Full is too powerful, so as to be undecidable and do not support the completeness check or efficient reasoning. The computational expense is huge if developing inference engine on OWL Full.

2) OWL DL

In order to enable efficient reasoning calculation, the sub-language OWL DL (Description Logic) of OWL Full limits the use of constructor of OWL and RDF. DL OWL with the above restrictions can be effectively inferred by the inference engine. We can acquire more knowledge by reasoning.

3) OWL lite

OWL Lite makes a further restriction on the basis of OWL DL: not allowed to use constructors like owl: one of, owl: disjoint with, owl: union of, owl: complement of, owl: has Value. Cardinal statement can only take 0 or 1, but not any non-negative integer. The statement owl: equivalent Class can only be used for class identifier, rather than anonymous class.

B. Ontology Construction Tools

Multiple tools are available for ontology construction, such as WebOnto, OpenCyc, OilEd, OntoEdit and Protégé, in which Protégé is an ontology development tool by the Biological Information Research Center of Stanford University. The latest version of Protégé, at present, is 5.1, which supports Chinese. It is based on Java language development, supporting RDF/RDFS, OWL DL and DL Query. Protégé is a software for ontology editing and knowledge acquisition, providing the construction of ontology concepts, classes, relationships, attributes and instances, as well as rich graphical user interface. In case of constructing ontology model at the conceptual level, Protégé allows developers to develop Plugin to expand its functionality [3]. On the whole, Protégé has been widely used in the field of ontology construction and research.

In the study herein, Protégé 5.1 is taken as the ontology construction tool, on the basis of referring to IDEF5 method, TOVE method, TUM method, Stanford seven-step method, cycling method and other ontology construction methods, to combine with the composition and characteristics of tax knowledge and borrow the software engineering software development life cycle method IEEE1074-2006][4], and NLP (natural language processing) and DLs (description logics) as the tool to construct the domain ontology of Chinese tax knowledge. Figure I. shows an example of the Chinese tax ontology constructed in Protégé.

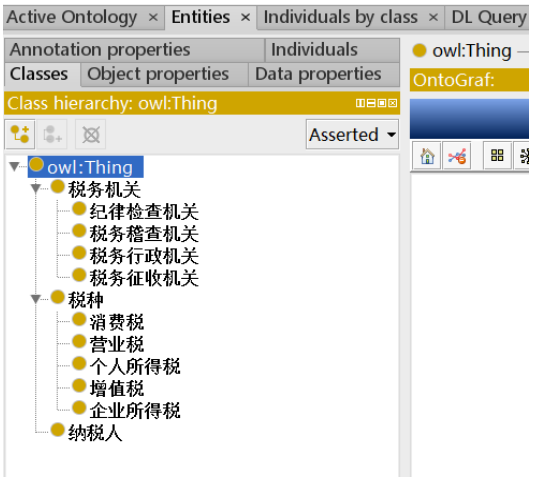

FIGURE I. CHINESE TAX ONTOLOGY CONSTRUCTED IN PROTÉGÉ

OntoGraf plug-in in Protégé can be used to draw ontology class hierarchy diagram, to facilitate our understanding of the concept hierarchy of ontology. This function is very useful both in the design construction phase of ontology, and in the use & maintenance phase. It enables us to quickly grasp the concept level of ontology, and then understand the semantic meaning of the domain ontology from the macro perspective.

III. ONTOLOGY PROGRAMMING TOOLS

Jena is an open-source Semantic Web development tool software package based on Java language [5]. With built-in inference engine, Jena supports RDF/RDFS, OWL and other ontology languages, and takes SPARQL as the ontology query language to use TDB or relational database to store ontology. Jena and Protégé are developed by Java language, which can work closely for use. Figure II. shows Jena framework architecture.

SPARQL (Simple Protocol and RDF Query Language) is an ontology query language defined by W3C that can be used for all information resources expressed by RDF. SPARQL, as an important technology of Semantic Web, has the potential to become the network database query language and data access standard.

The following is an example of the application code of Jena in China tax domain ontology:

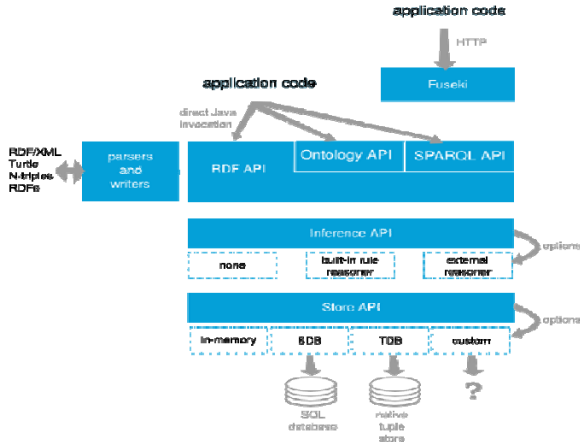

FIGURE II. JENA FRAMEWORK ARCHITECTURE [5]

IV. NATURAL LANGUAGE PROCESSING

Text classification is one of the main applications of natural language processing. Good text classification can greatly improve the efficiency of text information retrieval. Today, the popular natural language processing tools include a set of Java-based natural language processing toolkit by Stanford University[6]; NLTK (Natural Language Toolkit) developed by Python language, whose latest version is 3.2.1, supporting Stanford Word Segmenter and fully supporting Unicode encoding; and natural language processing toolkit SnowNLP for Chinese. These natural language processing tools have a rich text classification function. As Chinese does not use spaces to separate each word, the reader should separate each word according to its meaning. Most natural language processing software is based on the word segmentation for processing analysis. English text is composed of separate words, so words are already separated. But for Chinese, word segmentation is needed first, and then other classification, retrieval, statistics, feature extraction and other processing are required. The popular Chinese word segmentation software toolkits, at present, include JIEBA, NLPIR/ICTCLAS2016 word segmentation from Chinese Academy of Sciences, Stanford Word Segmenter and THULAC from Tsinghua University [8]. SnowNLP Chinese language processing toolkit has built-in Chinese word segmentation tool, applying Python language [9].

Natural language processing tools mentioned above provide the function of machine learning, so as to enable word segmentation, part-of-speech tagging, text classification, extraction of keywords, extraction of abstract and text similarity analysis, and support N-gram, HMM, Naïve Bayes, TextRank and other algorithms. In addition to rich text classification algorithm, NLTK in these tools support scikit-learn machine learning algorithm package [10]. In this paper, NLTK is mainly applied to study the text classification method based on machine learning and domain knowledge ontology.

V. REALIZATION OF ONTOLOGY TEXT CLASSIFICATION

The specific process of text classification in NLTK refers to Figure 5. The text feature is extracted from the input with feature extractor, such as statistical word frequency, grammatical features, keywords, and the classification model is trained by machine learning algorithm. With this model, classification is carried out for the text that needs to be classified. There are some problems with this method. For example, the frequency of some words is not high, but such words are very important; and each case appears summary words like "together", “total”, generally once, but the number right behind these words is the total number of the whole case. However, the total number is essential in the classification and query of the tax assessment cases. The method of feature extraction like statistical word frequency, grammatical features and keywords does not take into account the semantics of the text. The application of ontology can be effective in the design of feature extractors, and the accuracy of text classification can be improved by adding the analysis on semantics.

The algorithm proposed herein is to use the domain ontology constructed by Protégé, read the text to be classified through Jena, and take the domain ontology as the feature extractor of NLTK classifier for basic processing, reasoning and analysis. The specific implementation is as shown in Figure III.

It should be noted that Jena is developed in Java language,

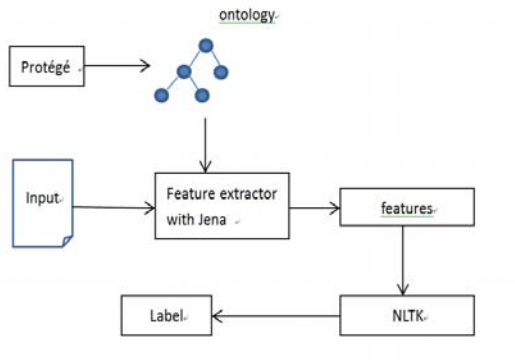

FIGURE III. SCHEMATIC DIAGRAM OF ALGORITHM STRUCTURE

and NLTK is developed based on Python. NLTK needs to access the ontology via Jena, and can be through Fuseki of Jena. Fuseki is a SPARQL server based on HTTP protocol. Python can query the ontology via a SPARQLWrapper package [11].

VI. TEST RESULTS

In this paper, Analysis Report on 100 Good Tax Assessment Cases of Chinese Tax System is mainly applied to study the text classification method based on machine learning and domain knowledge ontology, with a total of 100 cases. It is aimed to classify these cases by the category of tax, such as income tax, value-added tax, business tax and consumption tax. These categories can be mixed; that is, a document may belong to more than two categories at the same time. At first, 70 cases were classified by the category of tax as a training set; and the other 30 cases as a test set. User dictionary and tax domain ontology are respectively used as the basis for constructing the feature extractor, and Naïve Bayes classification algorithm module provided by NLTK is respectively trained and tested.

The three indicators tested are precision, recall and F-Score [12].

On the test set, TP (True Positive) indicates correct classification, that is, the results obtained by the classifier are consistent with the artificial classification; FP (False Positive) represents classification error, namely the results obtained by the classifier are inconsistent with the artificial classification; and FN (False Negative) represents the omitted cases that should be classified by the classifier.

Precision = TP/(TP+FP), namely the number of predicted true positive cases divided by the number of samples of all true positive examples;

Recall = TP/(TP+FN), namely the number of predicted true positive cases divided by the number of samples of all predicted true positive examples;

F-score = Harmonic average of precision and recall (2 × Precision × Recall)/(Precision +Recall).

Table I. shows the result of classification test by the classifier using user dictionary.

TABLE I. CLASSIFICATION RESULTS USING USER DICTIONARY

Category	TP	FP	FN	Precision	Recall	F-Score
Value-added tax	14	6	2	0.70	0.54	0.61
Business tax	6	2	3	0.75	0.46	0.57
Corporate income tax	12	7	2	0.63	0.86	0.36
Consumption tax	6	3	3	0.48	0.63	0.55

Table II. shows the result of classification test by the classifier using domain ontology.

TABLE II. CLASSIFICATION RESULTS USING DOMAIN ONTOLOGY

Category	TP	FP	FN	Precision	Recall	F-Score
Value-added tax	17	3	2	0.85	0.89	0.87
Business tax	7	1	3	0.88	0.70	0.78
Corporate income tax	15	4	2	0.79	0.88	0.83
Consumption tax	7	2	3	0.78	0.70	0.74

VII. SUMMARY

Through the comparison of test results, we can see a higher F-score for the classifier using domain ontology, which indicates better classification effect when the feature extractor using domain ontology considers the connotative semantics of the text. The domain ontology is taken as the basis of extraction algorithm of text feature extractor herein, and in addition, text features are considered from the semantic perspective, which possesses better performance as compared to the method only considering the keywords and word frequency relative to the user dictionary. During the research, tiny problems, like immature top ontology and smaller quantity of training set and test set, still exist, so further improvement is needed in the future research. In natural language processing, more and more research has noticed that the semantics of text cannot be ignored, and ontology, as a powerful tool for the current semantic description, shall be a key research focus in the future.

ACKNOWLEDGMENT

Supported by the National Natural Science Foundation of China (Grant No.71403151).

REFERENCES

[1] Beners-Lee T., H.J.L., The semantic web. Scientific American, 2001. 284(5): p. 34-43.

[2] Fürber, C., Semantic Technologies. 2016, Springer. p. 56-68.

[3] Musen, M.A., The Protégé project: A look back and a look forward, in Association of Computing Machinery Specific Interest Group in Artificial Intelligence. 2015, AI Matters.

[4] IEEE, IEEE Standard for Developing Software Life Cycle Processes, in IEEE1074-2006. 2006.

[5] Jena, A., A free and open source Java framework for building Semantic Web and Linked Data applications. Available online: jena. apache. org/(accessed on 28 April 2015), 2015.

[6] Manning, C.D., et al., The {Stanford} {CoreNLP} Natural Language Processing Toolkit. 2014. p. 55--60.

[7] Wagner, W., Natural Language Processing with Python, Analyzing Text with the Natural Language Toolkit by Steven Bird; Ewan Klein; Edward Loper. Language Resources & Evaluation, 2010. 44(4): p. 421-424.

[8] Zhongguo, L. and S. Maosong, Punctuation as Implicit Annotations for Chinese Word Segmentation. Computational Linguistics, 2009. 35(4): p. 205-512.

[9] Navathe, S.B., et al., Expert Recommendation in Time-Sensitive Online Shopping; Expert Recommendation in Time-Sensitive Online Shopping, S.B. Navathe, et al., S.B. Navathe, et al.^Editors. 2016. p. 297-312.

[10] Burghardt, M., et al. WebNLP – An Integrated Web-Interface for Python NLTK and Voyant. in Konvens. 2014.

[11] http://rdflib.github.io/sparqlwrapper/. 2016.

[12] James Pustejovsky, A.S., Natural Language Annotation for Machine Learning. 1st ed. 2013, Sebastopol: O'Reilly Media. 248.