TERM MATCHING APPROACHES IN E-COMMERCE

Mariam Gawich, Marco Alfonse, Mostafa Aref, Abdel-Badeeh M. Salem

***Abstract**: Information of a particular domain is presented in different terms with different structures. Terms can be found in unstructured text, semi structured forms (e.g. HTML and XML) and structured forms (e.g. databases and ontology). Computer can understand the semantic of a term from its relations with other terms. Term matching approaches are applied to detect the similar terms which yields to discover and present new integrated information. This paper investigates the term matching approaches applied in e-commerce.*

***Keywords**: Term Matching, String Similarity, Ontology Mapping, E-Commerce.*

***ACM Classification Keywords**: F.2.2 Nonnumerical Algorithms and Problems - Pattern matching*

Introduction

Term matching (TM) [Strzalkowski,1999] (also called term conflation) is considered as a field of text processing which belongs to information retrieval system. The objective of term matching is to cluster textual fragments in a document to similar terms or concepts. In addition, term matching is considered as a component of text analysis [Staab & Studer,2009] which is a stage of the process of information extraction. Term matching types are classified into syntactic matching and semantic matching. Syntactic matching [Giunchiglia & Shvaiko,2003] focuses on the string similarity between terms based on their form and it involves the syntax driven approaches. Semantic matching concerns about the interpretation of term meaning. The meaning of a term can be detected by its association with external knowledge sources such as semantic network, lexicon and ontology.

E-commerce is generally defined as the process of buying and selling goods and services on the internet. If there are two organizations want to make a deal on the internet, they will face a difficulty to understand each other if they don't have a homogeneous terminology. The matching between terms can solve this problem. The objective of this paper is the investigation of term matching approaches applied in e-commerce. This paper is organized as follows; section 1 introduces the term matching approaches in e-commerce, section 2 presents a framework proposal for comparative study and the last section contains the conclusion.

1. E-Commerce Term Matching Approaches

There are two main approaches for term matching in e-commerce which are the automated approach to product taxonomy mapping and the LexOnt matching approach. This section discusses in details the two approaches.

1.1 Automated Approach to Product Taxonomy Mapping in E-Commerce.

Lennart Nederstigt and colleagues proposed this approach [Nederstigt et al ,2014] to match two heterogeneous taxonomies provided by two ontologies. The algorithm receives two inputs; the first input is category taxonomy and its paths of the source taxonomy (the path is a list of nodes that start from root and end to the current node), while the second input involves the categories of target taxonomy. The approach was executed following these stages:

- **Preprocessing of category name**

The algorithm decomposes the name of a category of the source taxonomy into ampersands, comma. The result of this preprocessing is a set of multiple terms called split term set.

- **Word sense disambiguation**

This approach is derived from Park and Kim approach [Park & Kim,2007]. Its objective is to identify the correct meaning of a term presented by a leaf node (node which doesn't have a child) in the source taxonomy. The algorithm uses the Wordnet [Miller,1995] to search about the term provided by the split term set. The approach executes a comparison between the hyponyms of the term provided by the Wordnet and the same term provided by the split term set provided by the source taxonomy. The result of this process is an extended term set that is composed of original term and its synonyms. In order to detect the correct closely meaning of each term, the algorithm compares the sense hierarchy (hypernym relations) of the term provided by Wordnet with all ancestor nodes (upper category nodes which have children nodes) of the term provided by the current node located in the taxonomy source. The result of this comparison is a set of matched lemmas. To measure the convenience between each upper category with all sense hierarchy provided by the set of matched lemma, a similarity function [Aanen et al,2015] is applied as shown in equation 1.

$$\text{HyperProximity}(t,S)= \begin{cases} \dfrac{1}{\min_{x \in C}(\text{dist}(x,l))} & if\ C \neq \emptyset \\ 0 & if\ C=\emptyset \end{cases} \quad (1)$$

Where dist refers to the number of edges (connection between node and another), t denotes the upper category (ancestor), S denotes the sense hierarchy, l denotes the leaf located in the sense hierarchy S, C is a set of matching lemmas and x is a matching lemma. The output of this function is the matched lemma which its hypernym in Wordnet has the shortest distance to the leaf of the sense hierarchy. The average of hyper proximity is calculated to determine the overall similarity between the source category path and sense hierarchy.

- **Candidate path identification**

In this stage, the extended term set is used to determine the candidate path of the target taxonomy to be mapped with the current source category. The algorithm matches the terms provided by the extended split term set with paths provided by the target taxonomy. If one term of the extended term set is found to be a substring of the examined category of target taxonomy, the category is considered as candidate path. The candidate path identification is derived from Park and Kim algorithm [Park & Kim,2007] which compares the root node of the target path with the extended term set. The difference between the Park and Kim algorithm and proposed algorithm is that the later if it detects a term that is a substring of the actual tested category, category will be considered as candidate path and the algorithm will check the children of this category. Moreover, the proposed algorithm splits the original category name to multiple sets if it is a composite category. The algorithm compares between the multiple extended term set and the actual tested category name. This comparison requires the matching between every extended term set with its extended split term set. The result of the comparison is a boolean value true, if a term is a substring of the actual tested category or false, if there is no term can be a substring of the actual tested category. The path of the target category will be considered as a candidate path if half of boolean values are true.

- **Aggregated path similarity score**

To determine the best candidate paths of target taxonomy that can be adapted to match with source paths, an aggregated similarity score is calculated for each candidate path. Its objective is to measure the adaptation between target candidate path and the source path. The aggregated function is based on the use of Park and Kim algorithm [Park & Kim,2007] and the parent mapping similarity presented by the proposed algorithm. The aggregated similarity score consists of the cooccurence and order consistency measure. The cooccurence similarity [Aanen et al,2015] is based on lexical matching to detect the overlap between the category of target candidate path and the category of the source path regardless of their nodes order. Cooccurence similarity consists of Levenshtein similarity [Levenshtein,1966] and Jaccard similarity [Jaccard, 1912] .Order consistency is calculated to detect the common nodes that share the same order in the taxonomy hierarchy. It consists of the common PrecRel and consistent functions. The common function adds the node that match the category name

of the source taxonomy or its synonyms with another node of the target taxonomy or its synonyms. The synonyms are provided by Wordnet. The precRel function takes each common node to generate binary associations that indicates the order of relation between nodes (precedence relation). The consistent function determines if both of the two categories which are located in the candidate target path and two categories which are located in the source path have the same precedence relation. If they have the same precedence relation, the result of this function will be 1 and if they don't have the same precedence relation, the result of this function will be 0.

Concerning the evaluation, the algorithm is compared to Anchor Prompt algorithm [Noy& Musen,2004] and Park and Kim algorithm [Park & Kim,2007] .They are tested by three real life datasets; the first is provided by Open directory Project (ODP)- dmoz.org that consists of 44.000 categories, the second dataset is provided by Amzaon.com which more than 9.500 different categories are chosen for the evaluation and the third dataset is provided by Overstack.com that consists of 1.000 categories.

Table 1 shows that the automated approach achieves the highest value of recall (83%) and F1 (66%) measure compared to anchor prompt and Park and Kim. For precision, automated approach has a precision value less than Park and Kim. The high value of recall indicates that the automated approach can work on composite categories.

Table 1.Average results per algorithm [Nederstigt et al ,2014]

Algorithm	Precision	Recall	F1-measure	Computation time(s)
Anchor-PROMPT	28.93%	16.69%	20.75%	0.47
Park and Kim	47.77%	25.18%	32.52%	4.99
Lennart Nederstigt	38.28%	83.66%	52.31%	20.71

1.2 LexOnt Matching Approach

The objective of LexOnt approach [Arabshian et al,2012] is the production of frequent and significant terms provided by the corpus of Programmable Web (PW) directory [Programmable Web,2015]. Frequent and significant terms reflect the general properties of service classes provided by the PW to be automatically classified in ontology. The corpus of PW directory contains API description. The

corpus is encoded as HTML format. LexOnt algorithm relies on the information provided by the HTML text which describes the APIs service and information provided by Wikipedia that describe the domain of the service. Moreover LexOnt uses the Wordnet [Miller,1995] to detect the synonyms of terms to produce top N list words and phrases which can be used to determine distinct features of the service. LexOnt provides a semi-automatic ontology construction.

LexOnt Approach is executed by several algorithms that are outlined in the following stages:

- **TF-IDF (Term Frequency- Inverse Document Frequency)**

TF-IDF is calculated to demonstrate the importance of term appeared in the corpus. TF [Salton,1983] is defined as the frequency of a term t appeared in the corpus while IDF is the inverse document frequency which can be defined using this expression: [log(N/(nj+1))+1] where N is the total number of document and nj is the document frequency of term (t).

- **Significant phrases**

A significant phrase is composed of two or more words that can be a clue that indicates the high level property of a service class. For example, in the service 'Advertising' significant terms are 'Mobile Advertising', 'Facebook Advertising', etc. Significant phrase is detected through two steps; the first step is the determination of collocation, terms that occur together and the second step is the selection of unique collocations. The Chi Square is computed in this phase on collocated words to show the comparison between the numbers of times that words in a phrase are appeared together and the number of times that words appear alone. LexOnt uses the Wikipedia, Wordnet and constructed ontology to cover the main concepts and properties for an API service. For example, for the 'Advertisng' category, LexOnt algorithm will generate 20 top words provided by Wikipedia page which are (advertising, marketing, brand, television, semiotics, advertisement, billboard, radio, product, bowl, sponsor, consumer, advertise, placement, super, logo, commercial, infomercial, message, promotion). In addition, LexOnt applies the use of Wordnet to find the synonyms and related terms for each term listed in top N words. Also, LexOnt applies the matching between terms which are located in the constructed ontology and the generated terms. If there exist matched terms, LexOnt algorithm will rank and label them to mark that they are already existed in the ontology.

Tables 2, 3 and 4 demonstrate the evaluation of LexOnt; table 2 demonstrates the calculation of precision and recall of TF-IDF terms and the generation of significant phrases, table 3 demonstrates a second evaluation that involves the calculation of percentage of terms provided by external knowledge base (Wikipedia, Wordnet and Ontology) and table 4 shows a third evaluation which finds the percentage calculation of matched terms. For the three evaluations, the (Advertising, Real State, Social, Travel and Utility) categories are chosen for LexOnt testing. The (Advertising, Real State)

categories are selected upon the number of API (average of 40 API), specifity (has Wikipedia page) and prior knowledge of the domain (the co-author of the ontology should have a background about the service domain) . The (Social, Travel) categories are selected according to the familiarity of ontology creator with the service domain. The (Utility) category is chosen according to its number of APIs (65 APIs) and has no matched Wikipedia page. The equal terms, similar terms and different terms are illustrated in table 4.

Table 2. Precision/Recall Stats [Arabshian et al,2012]

Category	Sig.Phrase	TF-IDF	Recall
Advertising	3.98%	2.77%	43.88%
Real Estate	1.02%	.92%	9.57%
Social	3.21%	2.8%	20.19%
Travel	1.96%	2.4%	30.91%
Utility	9.58%	3.83%	34.91%

Table 3.Percentage of terms used from KB [Arabshian et al,2012]

Category	Sig.Phrase	TF-IDF
Advertising	41.38%	52.73%
Real Estate	100%	100%
Social	31.90%	11.38%
Travel	82.26%	72.73%
Utility	0%	0%

Table 4. Term Usage [Arabshian et al,2012]

Category	Equal Terms	Similar Terms	Different terms
Advertising	85.71%	100%	65.71%
Real Estate	16.67%	91.67%	66.67%
Social	1.73%	86.9%	79.1%
Travel	6.25%	100%	2.3%
Utility	5%	60%	50%

2. Framework Proposal for Comparative Study

The points of comparison that are used to point out differences between term matching approaches are; input, matching approach type, evaluation and output. Table 5 demonstrates the comparison between approaches.

Table 5. A Comparison betweeen Term Matching Approaches

Approach Name	Input	Matching Approach Type	Evaluation	Output
Automated approach to product taxonomy Mapping	Two ontologies	Semantic technique, statistical measure and substring technique	Yes	Mapping terms and candidate paths
LexOnt	Corpus, Wikipedia page and Ontology	String based technique and semantic matching	Yes	Significant terms and candidate terms for the ontology enrichment

First for the automated approach to product taxonomy mapping in e-commerce, its inputs are two ontologies in the domain of interest. The implementation of matching relies on several techniques; the semantic technique applied by the use of Wordnet and syntactic techniques that involve substring approach that detects the candidate path between target taxonomy and current source. The approach evaluation is based on its comparison with anchor prompt and Park and Kim algorithm. Second for LexOnt, its inputs are a corpus provided by the PW directory, Wikipedia page and constructed ontology in the domain of interest. The string matching approach is executed between word provided by API and top Wikipedia word. The semantic matching is applied through the Wordnet to detect the synonym of term.

Conclusion

Both of automated approach to product taxonomy mapping in e-commerce and LexOnt approach rely on the syntatic matching that focus on the structure of the word. Moreover, both of them execute the semantic matching by the use of Wordnet. Only LexOnt executes the mapping that focuses on the significant terms and suggests these to enrich the ontology which is considered as a technique for knowledge representation that can be used by other tools.

Bibliography

[Aanen et al,2015] SS.Aanen, D.Vandic and F. Frasincar. Automated product taxonomy mapping in an e-commerce environment. Expert Systems with Applications 42.3 (2015): 1298-1313, 2015.

[Arabshian et al,2012] K.Arabshian, P.Danielsenand and S.Afroz. LexOnt: A Semi-Automatic Ontology Creation Tool for Programmable Web. In: AAAI Spring Symposium: Intelligent Web Services Meet Social Computing. 2012.

[Giunchiglia & Shvaiko,2003] F.Giunchiglia and P. Shvaiko. Semantic matching. April 2003 Technical Report # DIT-03-0. University of Trento.Italy,2003.

[Jaccard, 1912] P.Jaccard. The distribution of the flora in the alpine zone." New phytologist 11.2, 1912.

[Levenshtein,1966] V. I. Levenshtein. Binary Codes Capable of Correcting Deletions, Insertions and Reversals." *Soviet Physics Doklady*. Vol. 10, 1966.

[Miller,1995] G.A.Miller. WordNet: A Lexical Database for English. Communications of the ACM 38(11), 39–41 .1995.

[Nederstigt et al ,2014] L.Nederstigt,D.Vandic and F.Frasincar. Automated Product Taxonomy Mapping In An E-Commerce. Erasmus University Rotterdam, 2014.

[Noy& Musen,2004] Natalya F. Noy and Mark A. Musen Using prompt ontology-comparison tools in the EON ontology alignment contest." Proceedings of the Third International Workshop Evaluation of Ontology-based Tools (EON). 2004

[Park & Kim,2007] S.Park and W.Kim. Ontology Mapping between Heterogeneous Product Taxonomies in an Electronic Commerce Environment. International Journal of Electronic Commerce 12(2), 69–87, 2007.

[Programmable Web,2015] Programmable Web, http://www.programmableweb.com/. (accessed: 4.05.2015).

[Salton,1983] G.Salton. Introduction to Modern Information Retrieval.McGraw-Hill, 1983.

[Staab & Studer,2009] S.Staab and R.Studer. Handbook On Ontologies.Part V- Ontology-Based Infrastructure and Methods.Page 663-666.Berlin: Springer, 2009. Print.

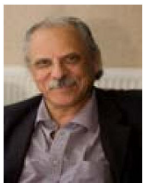
[Strzalkowski,1999] T. Strzalkowski. Natural Language Information Retrieval. NLP For Term Variant Extraction: Snergy Between Morphology, Lexicon And Syntax,1999.

Authors' Information

Mariam Gawich *is a PhD student* at the Faculty of Computer and Information Science, Ain Shams University, Cairo, Egypt.

Dr.Marco Alfonse Tawfik *is a Lecturer at the Faculty of Computer and Information Science, Ain Shams University, Cairo, Egypt. He got Ph.D. of Computer Science since August 2014, University of Ain Shams. His research interests: Semantic Web, Ontological Engineering, Medical Informatics, and Artificial Intelligence. He has published around 20 publications in refereed international journals and conferences.*

Prof. Mostafa M. Aref *is Professor of Computer Science, Ain Shams University, Cairo Egypt. He got Ph.D. of Engineering Science, June 1988, University of Toledo, Toledo, Ohio, USA. He has more than 50 journal and conference publications. His research areas are Natural Language Processing, Knowledge Representation, Object-oriented Programming, Ontology and Real-Time Strategy Games.*



Prof. Dr. Abdel-Badeeh M Salem *He is a Professor of Computer Science since 1989 at Ain Shams University, Egypt. His research includes intelligent computing, knowledge-based systems, biomedical informatics, and intelligent e-leaming. He has published around 250 papers in refereed journals and conferences. He has been involved in more than 400 Confs and workshops as a Keynote Speaker, Scientific Program Committee, Organizer and Session Chair. He is a member of many national and international informatics associations.*