A Comparison of Unsupervised Taxonomical Relationship Induction Approaches for Building Ontology in RDF Resources

Nansu Zong, Sungin Lee, and Hong-Gee Kim(✉)

Biomedical Knowledge Engineering Lab, School of Dentistry,
Seoul National University, Seoul, Korea
{zongnansu1982,sunginlee,hgkim}@snu.ac.kr

Abstract. Automatically generated ontology can describe the relationship of meta-data in Linked Data or other RDF resources generated from programs, and advances the utility of the data sets. Hierarchical document clustering methods used to generate concept hierarchies from retrieved documents or social tags can be used for constructing taxonomy or ontology for Linked Data and RDF documents. This paper introduces a framework for building an ontology using the hierarchical document clustering methods and compares the performance of three classic algorithms that are UPGMA, Subsumption, and EXT for building the ontology. The experiment shows EXT is the best algorithm to build the ontology for RDF resources and demonstrates that the quality of the ontology generated can be affected by the number of concepts that are used to represent the entities and to formalize the classes in the ontology.

Keywords: Taxonomical relationship induction · Hierarchy generation · Ontology generation · RDF · Linked data

1 Introduction

The popularity of RDF resources has grown gradually since 2001. For example, Linked Data increased in the number to 32 billion RDF triples by 2011[1]. The growing needs of the RDF resources push organizations to publish their own RDF format data, and they create RDF data by transforming their legacy data, such as relational database (RDB) or Web pages, by using transformation programs [5,11,18,30]. However, the use of the RDF data sets automatically generated from the programs has decreased due to lack of an ontology describing the relationship resident in the meta-data. For example, Linked Life Data[2] realizes mappings at both instance and predicate levels, but not at the class level because of lack of an ontology or concept hierarchy[3].

[1] http://events.linkeddata.org/ldow2012/
[2] http://linkedlifedata.com/
[3] http://linkedlifedata.com/sources

W. Kim et al. (Eds.): JIST 2013, LNCS 8388, pp. 445–459, 2014.
DOI: 10.1007/978-3-319-06826-8_32,

There are three ways for solving the problem: (1) generate an ontology from the data sources before publishing RDF data [1,26,35,36]; (2) map entities to existing ontologies during RDF data generation [3,28]; and (3) generate an ontology directly from RDF resources or Linked Data sets [24,40], a task similar to generating a concept hierarchy from retrieved documents or social tags in Information Retrieval (IR) known as hierarchical document clustering [31,33]. Linked Data sets and RDF resources store all the triples for entities as RDF documents [4]. This makes hierarchical document clustering methods potentially applicable to ontology generation in the Semantic Web.

In this paper, we introduce a framework for generating ontology from Linked Data or other RDF resources, based on hierarchical document clustering algorithms. We adapted the three most popular methods known as UPGMA [20], Subsumption [31], and EXT [19], to generate ontology in our framework. We evaluate the three algorithms with a preliminary experiment and the experiment shows that EXT is the best algorithm to build the ontology for RDF resources. We also notice that the quality of the ontology generated can be affected by the number of concepts that are used to represent the entities and to formalize the classes in the ontology, and suggest to improve the quality by changing the parameters of the algorithms to adjust the number of the layers in the generated ontology.

The rest of the paper is organized as follows: Sect. 2 introduces the related works; Sect. 3 introduces the framework designed for ontology generation; Sect. 4 systematically describes the methods using for the framework; Sect. 5 presents our experiment results; and finally, we provide conclusions in Sect. 6.

2 Related Works

The ontology generation or the concept hierarchy generation is considered as one branch of Knowledge Discovering and can be used for describing the relationship of meta-data [16,23]. Previous several studies used clustering and classification algorithms to build the taxonomical relationships based on inter-correlation attributes for databases [13,17]. Later, as the popularity of the Semantic Web grew, more studies used other machine learning approaches to build ontology, specially taxonomical relationship [7,32] to help transform databases into RDF data or to boost the application of the Semantic Web.

The taxonomical relationships or the concept hierarchy extracted from text data has been used to represent the search results in a hierarchy. The most traditional methods for building the hierarchy are based on hierarchical clustering algorithms known as agglomerative UPGMA and bisecting k-means [20]. The bisecting k-means is considered a better solution than UPGMA in performance [34]. Probability models are used to build the hierarchy and are reported as better algorithms than traditional ones in performance. The Subsumption [31], which is a kind of co-occurrence relationship of two concepts extracted from documents, is used to organize the concept hierarchy for retrieved documents. The subsumption is a kind of probability method to calculate the probability of

is_A relationship of two concepts and is considered as one of the most classical methods for concept hierarchy generation. Studies, such as [9,33], improved the subsumption-based approaches for different usages. Other studies are inspired to advance the precision of the subsumption-based method using a probability model. The DSP [22] computes the importance of every concept by topicality and predictiveness. The most important concepts are put to the top level of the hierarchy using the greedy approximation of the Dominating Set Problem (DSP). DisCover [21] maximizes hierarchy coverage while maintaining distinctiveness of concepts to identify the concepts in the hierarchy and receives better precision than DSP. FIHC [14] measures the cohesiveness of a cluster by using the frequent item sets that are calculated by the Global support and the Cluster support. FIHC is reported as better than agglomerative UPGMA and bisecting k-means [34]. Other famous studies that apply Formal Concept Analysis (FCA) to build a hierarchy are introduced in [8,12]. These studies build the concept-feature lattices and remove the features of the formalized graph to make a human readable hierarchy. Reference [25] uses Self-Organizing Map (SOM) to build the hierarchy by three different feature extraction approaches.

The concept hierarchy and the taxonomical relationships are also studied for social tags. Reference [33] uses Subsumption to induce the hierarchy from flicker tags. Reference [37] builds the hierarchy by using heuristic rules and deep syntactic analysis. EXT [19] induces a similarity graph to build the hierarchical taxonomy. An extensible greedy algorithm is used to place concepts that are the center of the similarity graph into the hierarchy. Reference [19] replaces the extensible greedy algorithm in [19] with a Directed Acyclic Graph (DAC) allocation algorithm, which allows the classes to maintain multiple super classes in the hierarchy.

In this research, we chose three most representative algorithms from the existing studies: a classic hierarchical clustering algorithm known as UPGMA [20,34], a popular probabilistic algorithm called Subsumption [9,31], and the most latest approach EXT [19].

3 Framework of Building Taxonomical Relationship of Entities

We separated the procedure of taxonomical relationship generation into four parts as shown in Fig. 1: Data Preparation, Pre-processing, Taxonomical Relationship Induction and Post-processing.

Data Preparation is designed to collect every piece of information about an entity to form an RDF document. We partitioned the RDF data graphs into small graphs called RDF documents, each of which contains the description of an entity. The triples or the quads are extracted to form a star-shaped document [4]. For example, an RDF document named “Acetylsalicylic acid” formalized by Data Preparation contains all the triples of the entity “Drugbank:drugs/DB00945” as the subject.

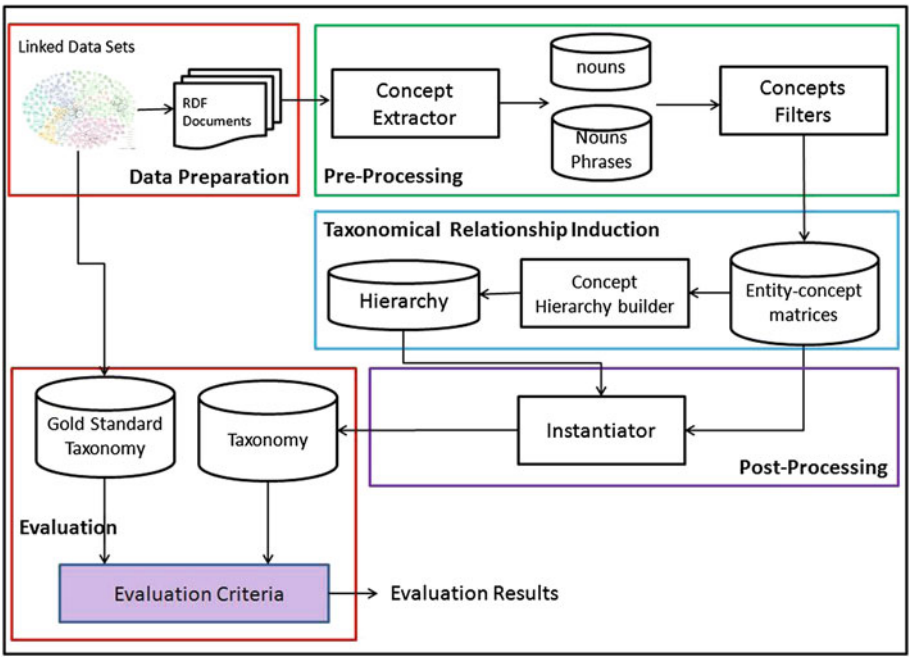


Fig. 1. The framework of building taxonomical relationship of entities

Pre-processing is designed to extract important nouns and noun phrases as candidate concepts of the RDF documents. First, the nouns and the noun phrases are extracted by a Part Of Speech (POS) tagger. Second, after lemmatizing and stemming, the nouns and the noun phrases are considered as candidate concepts in building the concept-entity matrix. Third, the concept-entity matrix is used to extract important concepts based on Vector Space Model (VSM) and Latent Semantic Analysis (LSA) [15]. The components of the matrix are calculated by Term Frequency and Inverted Document Frequency (TF-IDF) [29], and the matrix is decomposed by using Singular Value Decomposition (SVD) [2]. The important concepts are extracted by decreasing the context dimension by using LSA, and the important concepts are used to generate the new concept-entity matrix after pre-processing.

Taxonomical Relationship Induction is designed to construct the hierarchy based on the concept-entity matrix from the Pre-processing step. We implemented UPGMA, Subsumption and EXT introduced in Sect. 2 to build the taxonomy.

Post-processing is designed to instantiate the concepts of the hierarchy created in Taxonomical Relationship Induction. The concepts in the hierarchy are considered as classes in the ontology and the documents of an entity containing the concepts are considered as the instances of the classes.

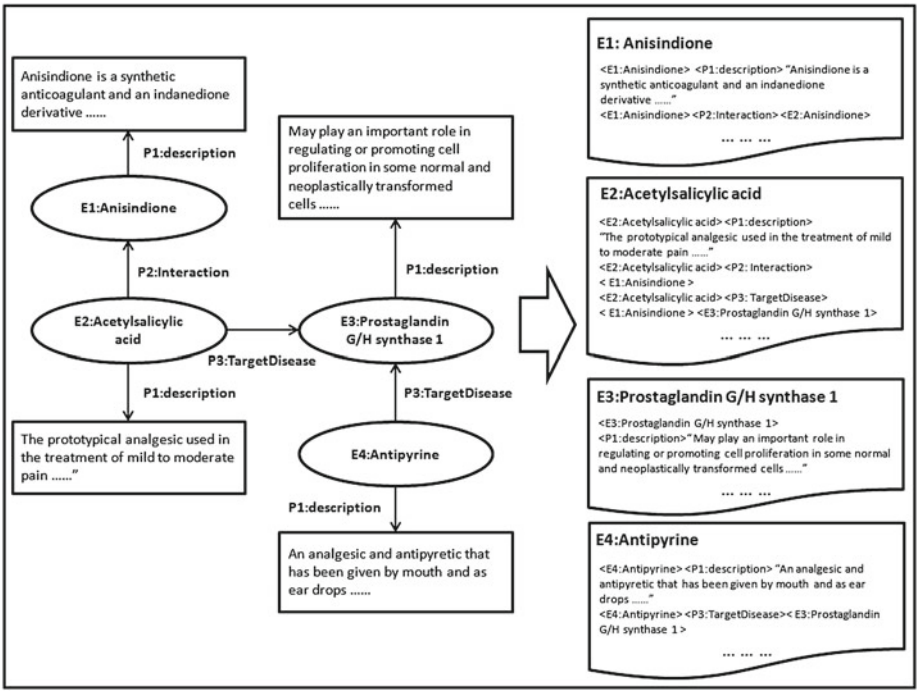


Fig. 2. An example of data preparation

4 Methodology

4.1 Data Preparation and Pre-processing

The Linked Data and other RDF resources are stored as either triples or RDF documents [6]. To adapt the existing hierarchical clustering algorithms, we need to process each entity as a document, which in effect partitions the whole data graph into small sub-RDF documents. For example, in Fig. 2, a data graph containing four entities is partitioned into four sub-RDF documents. We extracted nouns and noun phrases from each RDF document using the POS tagger after removing the punctuation marks and stop-words as mentioned in papers [27,39]. The nouns and the noun phrases tagged with NN (singular noun), NNP (proper noun), NNS (plural noun) and NNPS (plural proper nouns) are put into noun sequences. We generated candidate concepts from the noun sequences by three types of n-gram: unigrams, bigrams and trigrams [38]. The candidate concepts are lemmatized and stemmed for building the concept-entity matrix, which is used for finding the most important concepts that represent the entity.

We adapted VSM to represent documents. Each RDF document of the collection is defined as a single multidimensional vector $\boldsymbol{d} = [w_1, w_2, \ldots, w_i]$, where each component w_i is a weight of the dimension i and each dimension reflects

a candidate concept. We used TF-IDF to weight each component in the $\boldsymbol{d}$. The TF-IDF is calculated as follows:

$$w_{i,d} = tf_{i,d} * log \frac{\#documents}{\#(documents\ containing\ concept\ i)} \tag{1}$$

where the $tf_{i,d}$ is the normalized frequency of concept i in document d.

A vector of the single multidimensional vectors $\boldsymbol{d}$ is used to build an entity-concept matrix that is transposed into a concept-entity matrix. We extracted important concepts based on LSA after decomposing the matrix using SVD. SVD [2] is considered as a data reduction method that approximates the original concept-entity matrix in a lower dimension. With the help of the dimension reduction, the concept vector can be represented by the value of a new context, that is, a lower dimension instead of the original dimension reflecting the number of the entities. The concept-entity matrix A can be broken down into three matrices: an orthogonal matrix known as U, a diagonal matrix known as S, and a transpose of the orthogonal matrix known as V. SVD can be presented as follows:

$$A = U * S * V^T \tag{2}$$

where the columns of U are orthogonal eigenvectors of AA^T, the columns of V are orthogonal eigenvectors of A^TA, and S is a diagonal matrix containing the square roots of eigenvalues from U or V in descending order. The U can be considered as the representation of the concepts in the context S. LSA [15] finds a low-rank k approximation A_k to the concept-entity matrix A by selecting the k largest singular values of S, the first k columns of U and V^T. Hence U_k, formed by the first k columns of U, can be regarded as the representation of the important concepts in top k [39]. In practice, we extracted the important concepts that reflect the most weighted values of the U_k in each column. Then the entity can be represented with the most important concepts extracted. The important concepts in top k and the entities form a new concept-entity matrix M. A simple example of the procedure of pre-processing is shown in Fig. 3.

4.2 Taxonomical Relationship Induction

In this section, we present how to construct a concept hierarchy for the three methods using the new concept-entity matrix M.

UPGMA. UPGMA [20] constructs a concept hierarchy based on the similarity of two concepts. The computation of the similarity of two concepts is based on VSM as shown below:

$$Sim(c_1, c_2) = \frac{\sum_1^n (w_{(e_i,c_1)} * W_{(e_i,c_2)})}{\sqrt{\sum_1^n (w_{(e_i,c_1)})^2} * \sqrt{\sum_1^n (w_{(e_i,c_2)})^2}} \tag{3}$$

where $w_{(e_i,c_j)}$ is the IF-IDF value of the entity e_i and the concept c_j computed by Eq. 1. The new node, the superclass of the two most similar concepts, is created

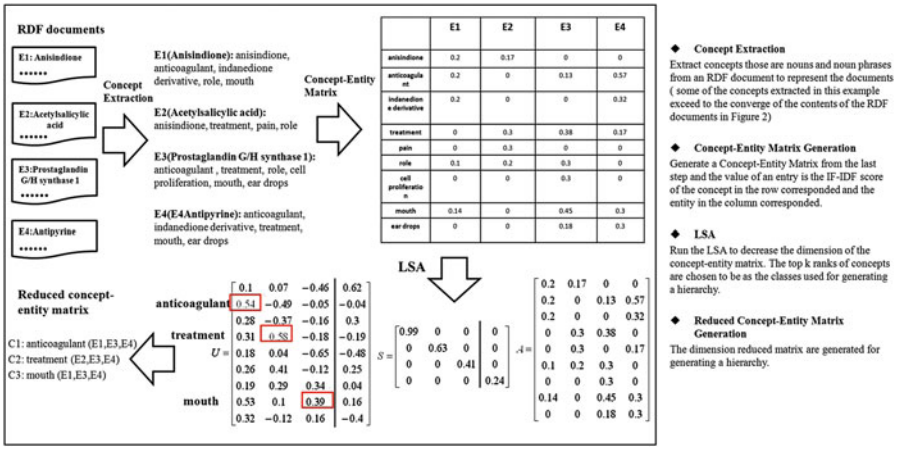


Fig. 3. An example of the procedure of the pre-processing

as a virtual class that has the common entities shared by the two concepts. We generate the concept hierarchy by using the Algorithm 1 as shown below:

Algorithm 1. UPGMA-based tree generation

Input: concept-entity matrix M and the index I of the RDF documents
Output: concept hierarchy

put all the concepts in Queue Q
put *root* into a hierarchy H
while the size of $Q > 0$ **do**
  for all each concept c_i in Q **do**
    for all each concept c_j in Q **do**
      compute the similarity of c_i and c_j using the Equation 3
    end for
  end for
  create new class node of the super class C_{i+j} of the most similar concepts c_i and c_j in H
end while

Subsumption. Subsumption [31] defines the is_A relationship based on the co-occurrences in different RDF documents of entities. Given $c1$ and $c2$, $c1$ is said to subsume $c2$ if the following conditions are satisfied:

$$P(c1|c2) > 0.8,\ P(c2|c1) < 1 \tag{4}$$

We generate the concept hierarchy using the Algorithm 2 shown as follows:

EXT. EXT [19] uses an extensible greedy algorithm to put the most important concept into the subclass of the most similar concept in the current level of the concept hierarchy. The sequence in which to choose the most important concept is decided by the centrality of the similarity graph created from the concept-entity matrix. The centrality of the concepts in the similarity graph is computed as follows:

Algorithm 2. Subsumption-based tree generation

Input: concept-entity matrix M and the index I of the RDF documents
Output: concept hierarchy

```
for all each concept c_i in M do
  for all each concept c_j in M do
    if c_i and c_j satisfies the condition in Equation 4 then
      put c_j to the subclass of c_i
    end if
  end for
end for
sort concepts by descending order of the number of subclass in Queue Q
put root into a hierarchy H
for all each concept c_i in Q do
  for all each subclass c_j do
    put the class node of c_j to the subclass of class node of c_i in H
  end for
end for
```

$$Cen(c_x) = \sum_{c_x \neq c_y \neq c_z \in C} \frac{\delta_{c_y c_z}(c_x)}{\delta_{c_y c_z}} \tag{5}$$

where $\delta_{c_y c_z}$ is the total number of shortest paths from concept c_y to concept c_z, and $\delta_{c_y c_z}(c_x)$ is the number of concepts that pass through c_x. The similarity graph consists of vertices that are concepts, and edges that are added if the similarity of the two concepts is above the threshold α. The similarity of the two concepts is calculated by using Eq. 3, and whether a concept c_i should be put as the subclass of a concept c_j depends on the threshold β. Concept hierarchy is generated by using Algorithm 3 as follows:

Algorithm 3. EXT-based tree generation

Input: concept-entity matrix M and the index I of the RDF documents, threshold1 α, and threshold2 β
Output: concept hierarchy

```
put all concepts into a similarity graph G
for all each concept c_i in M do
  for all each concept c_j in M do
    if the similarity of c_i and c_j is above α then
      put an edge connecting c_i with c_j into G
    end if
  end for
end for
sort concepts by descending order of centrality computed by Equation 5 in Queue Q
put root into a hierarchy H
while the size of Q > 0 do
  for all each concept c_i in Q do
    for all each class c_j in H do
      compute the similarity of c_i and c_j
    end for
    if the most similarity of the most similar class c_j of the c_i is above β then
      put the c_i as the subclass of c_i in H
    else
      put the c_i as the subclass of root in H
    end if
  end for
end while
```

4.3 Post-processing

We put all the entities into the generated concept hierarchy based on the concept-entity matrix. For example, a class named "anticoagulant" in Fig. 3 will be assigned with three instances "E1:anisindione", "E3:porstaglandin G/H synthase 1", and "antipyrine". Notice that, since UPGMA and Subsumption support multiple inheritance [31,34], we also allow multiple inheritance for the assignment of instances during the post-processing step.

5 Preliminary Experiment

We implemented our framework based on JDK_1.6 using the Intel, I-5 CPU with 4 GB RAM and 1TB hard disk on a Ubuntu[4] system.

5.1 Data Set and Gold standard

We used Diseasome[5] from Linked Life Data[6] as our source data for our experiment. The Diseasome supplies entities about diseases and also describes schemas relationships. We separated the data set into two parts: data that contains the triples of entities, and a gold standard that contains the triples about the schema. For example, given two triples "$< Diseasome : diseases/272 >< Diseasome : class >< Diseasome : diseaseClass/Endocrine >$" and "$< Diseasome : diseases/2013 >< Diseasome : subtypeOf >< Diseasome : diseases/272 >$", we can get an is_A relationship in which a class named "Endocrine" has a subclass named "diseases:272" with an instance named "diseases:2013". The Diseasome contains 1308 instances of diseases, and we used them to create a disease ontology in our experiment.

5.2 Data Preparation and Pre-Processing

We partitioned the data into star-shaped RDF documents and parsed the documents using the POS tagger[7]. The nouns and noun phrases extracted were stemmed and lemmatized by the Lucene String Parser[8] to form a concept-entity matrix. We used JAMA[9] to run SVDs to choose important concepts for the entities. The SVDs are computed with the different values of k such as "100, 300 and 500", each of which means the number of concepts extracted to build an ontology. The three different concept-entity matrices, dimensionally reduced, cover different entities of the original data set, affecting the recall for the constructed ontology. Figure 4 shows the coverage of the entities by the different values of k.

[4] http://www.ubuntu.com/
[5] http://diseasome.eu/
[6] http://linkedlifedata.com/
[7] http://nlp.stanford.edu/software/tagger.shtml
[8] http://lucene.apache.org/
[9] http://math.nist.gov/javanumerics/jama/

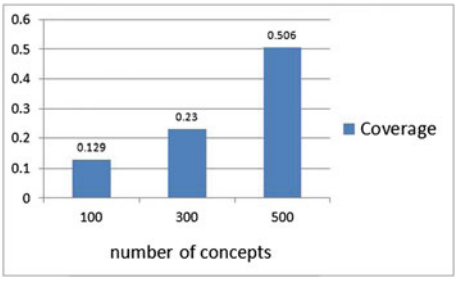


Fig. 4. The coverage of using different number of concepts for building an ontology

The coverage is computed by $coverage(k) = \frac{\#entities(k)}{\#entities}$, where $entities(k)$ is the entities remained by reducing the dimension into k. Using 100 concepts only covers 13 % of the entities and using 500 51 % of the entities. We noticed that coverage increases as the number of concepts increases.

5.3 Criteria for Evaluation

In order to evaluate the taxonomies generated by the three approaches, we compared generated ontologies with the gold standard. Since class names in the taxonomies vary in the three methods, lexical comparisons of the class names are inappropriate. We adapted the Taxonomic Precision (TP) and Taxonomic Recall (TR) [10,25] to measure the quality of the generated ontologies.

TP and TR are based on the Semantic Cotopy (SC), which defines super-sub concepts (is-A) relations in an ontology:

$$SC(c, o) = \{c_i | c_i \in C \wedge (c_i \leqslant c \vee c \geqslant c_i)\} \tag{6}$$

The Common Semantic Cotopy (CSC) that avoids the influence of lexical precision in the taxonomic measurement can be defined as:

$$CSC(c_i, O_1, O_2) = \{c_j | c_i \in C_1 \cap C_2 \wedge (c_j \leqslant C_1 c_i \vee c \geqslant C_1 c_j)\} \tag{7}$$

The TP_{CSC} and TR_{CSC} can be computed as:

$$TP_{CSC}(O_1, O_2) = \frac{1}{|CO_1 \cap CO_2|} \sum_{c \in CO_1 \cap CO_2} TP_{CSC}(c, c, O_1, O_2) \tag{8}$$

$$TR_{CSC}(O_1, O_2) = TP_{CSC}(O_2, O_1) \tag{9}$$

Taxonomic F-measure (TF) calculates the harmonic mean of TP_{CSC} and TR_{CSC}:

$$TF(O_1, O_2) = \frac{2 * TR_{CSC}(O_1, O_2) * TP_{CSC}(O_1, O_2)}{TR_{CSC}(O_1, O_2) + TP_{CSC}(O_1, O_2)} \tag{10}$$

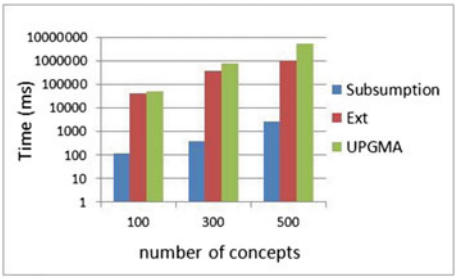


Fig. 5. The running time of the three algorithms

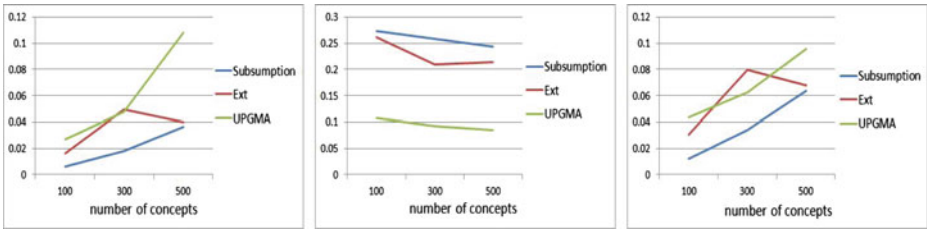
5.4 Results

We used UPGMA, Subsumption, and EXT to generate ontologies in our experiment. For EXT, the α and the β are both set to 0.5.

Figure 5 shows the running times of the three algorithms. Subsumption is the fastest algorithm: 116 ms for 100 concepts, 391 ms for 300, and 2683 ms for 500. The running times of the three methods increase as the number of concepts used increases.

Figure 6 shows ontology quality as the number of concepts involved changes. For TR, UPGMA gets the highest score among the three algorithms: 11 % for 500 concepts. The low TRs obtained by the algorithms are in part due to the decomposition of the concept-entity matrix, which makes the converge of entities for building the ontologies limited. For TP, Subsumption and EXT show almost the same precision that reaches 24 % when using 500 concepts. The F-measure demonstrates that quality increases as more concepts are used.

We used only those entities of the gold standard that were covered by the new decomposed concept-entity matrix to get the F-measure shown in Fig. 7. The figure shows that UPGMA performs poorer than the other two algorithms on F-measure but performs the best on Recall. EXT performs better than both Subsumption and UPGMA on F-measure. Subsumption received the best score



(a) The taxonomic recall (b) The taxonomic precision (c) The F-measure

Fig. 6. The quality of the ontologies generated by the three algorithms

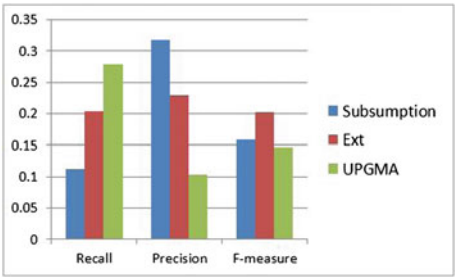


Fig. 7. The average TP, TR, and F-measure of the ontologies generated by the three algorithms, which are evaluated by the sub-structure in the gold standard. The sub-structure only contains the entities used after reducing dimensions of RDF documents

on Precision. Considering the running times of the three algorithms, the authors regarded EXT as the best algorithm for our purpose.

Each hierarchical document clustering method adapted in this paper builds an ontology that has more layers than the ontology generated from the gold standard has, affecting the quality of the former ontology. However, the number of layers in the ontology can be reduced by adjusting parameters in the algorithms - an improvement work reserved for the future.

6 Conclusion

In this paper, we introduced a framework for generating ontology from Linked Data or RDF resources based on hierarchical document clustering algorithms that are used to generate concept hierarchies from retrieved documents or social tags. We implemented three most classic and popular methods for hierarchical document clustering to generate ontology in our framework. We evaluated the three algorithms with a preliminary experiment and the experiment shows EXT is the best algorithm to build the ontology for RDF resources. We also learned that the quality of the ontology generated can be affected by the number of concepts that are used to represent the entities and to formalize the classes in the ontology, and suggested an ontology quality improvement that reduces the number of layers in the ontology.

Acknowledgments. This research was funded by the MSIP(Ministry of Science, ICT & Future Planning), Korea in the ICT R&D Program 2013. We appreciate Ashley Hess for the proof reading.

References

1. Alani, H., Kim, S., Millard, D.E., Weal, M.J., Hall, W., Lewis, P.H., Shadbolt, N.R.: Automatic ontology-based knowledge extraction from web documents. IEEE Intell. Syst. **18**(1), 14–21 (2003)

2. Baker, K.: Singular Value Decomposition Tutorial. The Ohio State University, Columbus (2005)
3. Berners-Lee, T., et al.: Linked data-the story so far. Int. J. Semant. Web Inf. Syst. **5**(3), 1–22 (2009)
4. Blanco, R., Mika, P., Vigna, S.: Effective and efficient entity search in RDF data. In: Aroyo, L., Welty, C., Alani, H., Taylor, J., Bernstein, A., Kagal, L., Noy, N., Blomqvist, E. (eds.) ISWC 2011, Part I. LNCS, vol. 7031, pp. 83–97. Springer, Heidelberg (2011)
5. Blum D., Cohen, S.: Generating RDF for application testing. In: ISWC Posters & Demos (2010)
6. Bron, M., Balog, K., de Rijke, M.: Example based entity search in the web of data. In: Serdyukov, P., Braslavski, P., Kuznetsov, S.O., Kamps, J., Rüger, S., Agichtein, E., Segalovich, I., Yilmaz, E. (eds.) ECIR 2013. LNCS, vol. 7814, pp. 392–403. Springer, Heidelberg (2013)
7. Cerbah, F.: Mining the content of relational databases to learn ontologies with deeper taxonomies. In: 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, pp. 553–557. IEEE (2008)
8. Cimiano, P., Hotho, A., Staab, S.: Learning concept hierarchies from text corpora using formal concept analysis. J. Artif. Intell. Res. (JAIR) **24**, 305–339 (2005)
9. De Knijff, J., Frasincar, F., Hogenboom, F.: Domain taxonomy learning from text: the subsumption method versus hierarchical clustering. Data Knowl. Eng. **83**, 54–69 (2013)
10. Dellschaft, K., Staab, S.: On how to perform a gold standard based evaluation of ontology learning. In: Cruz, I., Decker, S., Allemang, D., Preist, C., Schwabe, D., Mika, P., Uschold, M., Aroyo, L.M. (eds.) ISWC 2006. LNCS, vol. 4273, pp. 228–241. Springer, Heidelberg (2006)
11. Ding, L., DiFranzo, D., Graves, A., Michaelis, J.R., Li, X., McGuinness, D.L., Hendler, J.A.: TWC data-gov corpus: incrementally generating linked government data from data. gov. In: Proceedings of the 19th International Conference on World Wide Web, pp. 1383–1386. ACM (2010)
12. Drymonas, E., Zervanou, K., Petrakis, E.G.M.: Unsupervised ontology acquisition from plain texts: the *OntoGain* system. In: Hopfe, ChJ, Rezgui, Y., Métais, E., Preece, A., Li, H. (eds.) NLDB 2010. LNCS, vol. 6177, pp. 277–287. Springer, Heidelberg (2010)
13. Fisher, D.: Improving inference through conceptual clustering. In: Proceedings of AAAI Conference 1987, pp. 461–465 (1987)
14. Fung, B.C.M., Wang, K., Ester, M.: Hierarchical document clustering using frequent itemsets. In: Proceedings of SIAM International Conference on Data Mining, pp. 59–70 (2003)
15. Garcia, E.: Latent semantic indexing (lsi) a fast track tutorial (2006)
16. Han, J., Cai, Y., Cercone, N.: Knowledge discovery in databases: an attribute-oriented approach. In: Proceedings of the International Conference on Very Large Data Bases, pp. 547–547. Citeseer (1992)
17. Han, J., Fu, Y.: Dynamic generation and refinement of concept hierarchies for knowledge discovery in databases. In: Proceedings of AAAI, vol. 94, pp. 157–168 (1994)
18. Heath, T., Bizer, C.: Linked data: evolving the web into a global data space. Synth. Lect. Semant. Web: Theory Technol. **1**(1), 1–136 (2011)
19. Heymann, P., Garcia-Molina, H.: Collaborative Creation of Communal Hierarchical Taxonomies in Social Tagging Systems. Stanford Infolab Publications, Heidelberg (2006)

20. Jain, A.K., Dubes, R.C.: Algorithms for Clustering Data. Prentice-Hall Inc., New Jersey (1988)
21. Kummamuru, K., Lotlikar, R., Roy, S., Singal, K., Krishnapuram, R.: A hierarchical monothetic document clustering algorithm for summarization and browsing search results. In: Proceedings of the 13th International Conference on World Wide Web, pp. 658–665. ACM (2004)
22. Lawrie, J.D., Croft, W.B.: Generating hierarchical summaries for web searches. In: Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval, pp. 457–458. ACM (2003)
23. Li, C.: Knowledge discovery in database. J. Northwest.Univ.: Nat. Sci. Ed. **29(1)**, 114–119 (1999)
24. Parundekar, R., Knoblock, C.A., Ambite, J.L.: Linking and building ontologies of linked data. In: Patel-Schneider, P.F., et al. (eds.) ISWC 2010. LNCS, vol. 6496, pp. 598–614. Springer, Heidelberg (2010)
25. Paukkeri, M.-S., García-Plaza, A.P., Fresno, V., Unanue, R.M., Honkela, T.: Learning a taxonomy from a set of text documents. Appl. Soft Comput. **12**(3), 1138–1148 (2012)
26. Pivk, A.: Automatic ontology generation from web tabular structures. AI Commun. **19**(1), 83–85 (2006)
27. Paolo-Ponzetto, S., Strube, M.: Taxonomy induction based on a collaboratively built knowledge repository. Artif. Intell. **175**(9), 1737–1756 (2011)
28. Sahoo, S.S., Halb, W., Hellmann, S., Idehen, K., Thibodeau, Jr., T., Auer, S., Sequeda, J., Ezzat, A.: A survey of current approaches for mapping of relational databases to RDF. W3C RDB2RDF Incubator Group Report (2009)
29. Salton, G., Buckley, C.: Term-weighting approaches in automatic text retrieval. Inf. Process. Manag. **24**(5), 513–523 (1988)
30. San Martín, M., Gutierrez, C.: Representing, querying and transforming social networks with RDF/SPARQL. In: Aroyo, L., et al. (eds.) ESWC 2009. LNCS, vol. 5554, pp. 293–307. Springer, Heidelberg (2009)
31. Sanderson, M., Croft, B.: Deriving concept hierarchies from text. In Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 206–213. ACM (1999)
32. Santoso, H.A., Haw, S.C., Abdul-Mehdi, Z., et al.: Ontology extraction from relational database: concept hierarchy as background knowledge. Knowl.-Based Syst. **61(8)**, 729–741 (2010)
33. Schmitz, P.: Inducing ontology from flickr tags. Collaborative Web Tagging Workshop at WWW2006 **50**, 210–214 (2006)
34. Steinbach, M., Karypis, G., Kumar, V., et al.: A comparison of document clustering techniques. In: KDD Workshop on Text Mining, pp. 525–526. Bibliometric, Boston (2000)
35. Tho, Q.T., Hui, S.C., Fong, A.C.M., Cao, T.H.: Automatic fuzzy ontology generation for semantic web. IEEE Trans. Knowl. Data Eng. **18(6)**, 842–856 (2006)
36. Tijerino, Y.A., Embley, D.W., Lonsdale, D.W., Ding, Y., Nagy, G.: Towards ontology generation from tables. World Wide Web **8**(3), 261–285 (2005)
37. Tsui, E., Wang, W.M., Cheung, C.F., Lau, A.S.M.: A concept-relationship acquisition and inference approach for hierarchical taxonomy construction from tags. Inf. Process. Manag. **46**(1), 44–57 (2010)
38. Yang, H.: Personalized concept hierarchy construction. Ph.D. thesis, University of Southern California (2011)

39. Zheng, H.-T., Borchert, C., Kim., H.-G.: A concept-driven automatic ontology generation approach for conceptualization of document corpora. In: IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, WI-IAT'08, vol. 1, pp. 352–358. IEEE (2008)
40. Zong, N., Im, D.-H., Yang, S., Namgoon, H., Kim, H.-G.: Dynamic generation of concepts hierarchies for knowledge discovering in bio-medical linked data sets. In: Proceedings of the 6th International Conference on Ubiquitous Information Management and Communication, p 12. ACM (2012)