Alignment of Large Web Taxonomies using Hierarchical Classification

Bharadwaj J
Department of Computer Science and Engineering
Indian Institute of Technology Madras

***Abstract*—Most websites organize their webpages, into a well defined hierarchy, known as web taxonomy. Different websites follow different taxonomies, even if they belong to the same domain. For a given domain, our objective is to align nodes or entities from different websites, and generate a common unified taxonomy for that domain.**

Keywords*—*Web Taxonomy, Hierarchical Classification,

I. Introduction

Taxonomies are a common way of organizing large amounts of information. Websites organize their web content into taxonomies, that allow easy access and retrieval of information. For example, if we visit an e-commerce website, we can select a product by navigating to the desired categories, instead of looking at every product available in the website. However, different websites from the same domain, use different taxonomies, thus making it difficult to collect information about a particular entity, that might be available in several websites of that domain. In the e-commerce example stated above, the same product might be listed under different categories in different websites, depending on the product taxonomy followed by that website. Hence, our aim is to generate a standard unified taxonomy for a given domain, that might be applicable to different websites under that domain. The problem of Taxonomy Alignment, has been studied in the past and there exist several approaches and tools for performing the same. To the best of our knowledge, this is the first approach, that uses a supervised learning-based approach for performing taxonomy alignment. The significance of such an approach is that, we can learn similarity between different entities, even if the data corresponding to the entities is available in a raw unprocessed form. In our approach, we use raw text data taken from webpages under different entities, to learn a Hierarchical Classifier for a particular taxonomy, and perform alignment based on the predictions made by this classifier. *Hierarchical Classification* [2], is a commonly used technique for classifying hierarchical data, where the input data points are mapped to categories, that form a hierarchical tree structure. It has been experimentally proved that, hierarchical classification performs better that flat classification for such data [7].

II. Related Work

The problem of aligning taxonomies has been investigated in the past and different kinds of approaches have been proposed. These approaches mainly depend on, what kind of data is available. In some cases, only the taxonomy entity names might be available. In other cases, the relationship between these entities, might also be known. In the case of web taxonomies, it is possible to collect web pages under each entity and use this data for alignment.

Alignment and Matching are usually done in two ways. In *schema-based matching* approaches, the hierarchical data is encoded as an XML schema or other similar formats, and schema matching is performed. Examples of such algorithms are Cupid[5], SemInt[6] and COMA[4]. These methods perform matching, purely based on the schema, or might make use of additional meta-information such as a linguistic corpus. *Instance-based matching* approaches, make use of the data under the different categories, and match them using some data-driven algorithm. One of the earliest instance based methods, was proposed by Philip Resnik[3], in which the similarity between nodes in taxonomies, was computed based on the information sharing between the nodes. Using this approach, one can compute the similarity between nodes within a given taxonomy. However, we need to compare nodes across taxonomies to perform alignment. In [8], database schemas are matched, using a query discovery process. However, this method might be applicable only when the data is in the form of a database with attributes, attribute values and attribute relationships. A detailed comparison of the two types of matching is done in [1].

We propose a novel approach, that uses raw unprocessed data and performs supervised learning-based alignment. The approach will perform an instance-based alignment of product taxonomies of different e-commerce websites, by learning a hierarchical classifier on product data.

III. Hierarchical Classification

Product pages in online shopping websites are organized into categories, based on a product taxonomy. Hence, we can train a hierarchical classifier on the product taxonomy. For each level in the taxonomy, we train a classifier for each category at that level, with the subcategories under that category as class labels. Thus, the number of classifiers to be trained, is equal to the number of internal nodes in the taxonomy. Once we have a hierarchical classifier in place, it is possible to predict the categories of products from from other websites. Hence, we need to collect sufficient amount of data from different websites, covering all possible categories of products, and construct a hierarchical classifier on each of these taxonomies. After this, a simple voting method can be used to align corresponding nodes of different taxonomies. We can consider a single taxonomy, T_B as the reference taxonomy and predict data collected from other taxonomies, T_i. For each category c_{ij} in T_i, predict the data under that category, using the hierarchical classifier trained on T_B. If a major percentage of data, belonging to category c_{ij} in taxonomy T_i, get mapped to category c_{Bk} in taxonomy T_B, then it indicates that category c_{Bk} in T_B, is similar to category c_{ij} in taxonomy T_i. Thus, the category c_{ij}, is mapped or aligned to that subcategory in T_B, that gets the maximum number of votes.

IV. Work Completed

Around 30 million product pages, covering a wide variety of categories from amazon.com were collected and, their product titles were extracted and *bread crumbs* were formed. A bread crumb indicates a branch in the product taxonomy, to which a product belongs. They are of the form: $cat_1 > cat_2 > cat_3 > ... > cat_n$, where cat_i is the category at the i^{th} level of that branch. However, the bread crumbs generated were malformed, and had to be preprocessed before using them as labels. Various issues like missing categories, incorrect category positions within the bread crumb, empty categories, broken bread crumbs were fixed during the preprocessing stage, which involved manual realignment with respect to the top level category. The other major challenge involved, was the presence of redundant and rare categories. There are certain categories, that contain products that might also be listed in a different category. For example, the category *Computer Accessories* contains several products that are listed in *Computers* category. In addition, there are some categories, that contain very few product titles under them, at various levels of the hierarchy. Currently, we are not taking such categories for training, as our approach requires large amounts of data under each category. Hence, a non-instance based approach can be applied at a later stage, for such categories alone.

Product titles alone, were used as data for training, as these are quite long and contain sufficient information about the category. Linear Support Vector Machines, were used for training, as the dimension of text can be significantly large, and Linear SVMs scale well under these conditions. LIBLINEAR library [9] was used for training the Linear SVMs. Training was performed level by level, with the i^{th} category in the bread crumb serving as the label for the i^{th} level classifier. The classifiers were trained, using k-fold Cross Validation, with the value of k being varied from level to level. At the top level, ten fold cross-validation was performed. As we go lower, the amount of data and the number of class labels is smaller in number, and hence the number of folds taken was between 3-5 for these levels. As of now, training has been completed for the top three levels. For most of the categories, this happens to be the last level. However, few categories like *electronics* are quite deep and have additional levels to be trained.

V. Results

The level wise results of Hierarchical Classification on Amazon data are given in Table I. The results indicate that, the performance improves as we go down the hierarchy. At lower levels the number of subcategories under each category is considerably smaller, and hence the performance becomes better. The average f_1 score indicated in Table I, is an average of the f_1-scores of each classifier in that level.

TABLE I. Hierarchical Classification: Level Wise Results

Level	No. of Classifiers	Average test f_1 - score
1	1	0.775
2	28	0.796
3	190	0.807

Fig. 1 shows the precision, recall and f_1-scores obtained for each individual top level category. Table II indicates the mapping between the categories and their labels displayed in Fig 1. The *Jewelry* class has the highest f_1 score, owing to its well-defined titles and high support, whereas the *Home Improvement* category has the least score. The other categories also follow a similar trend, with a few exceptions like *Clothing, Baby, Kitchen & Dining,* etc. This might be, due to the varying similarity among the subcategories under these categories, and will get rectified, as we go down the hierarchy.

Fig. 1. Performance of top level classifier

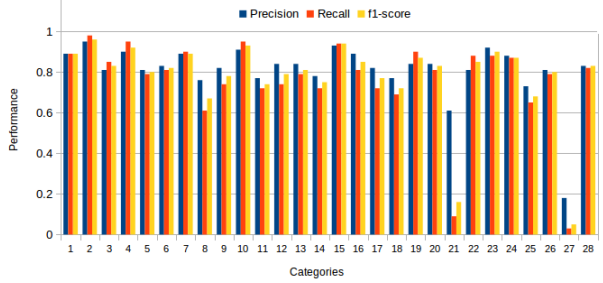
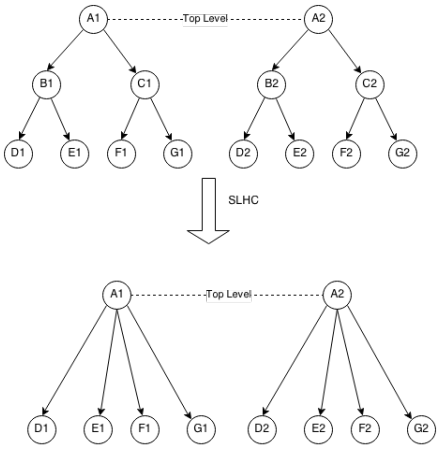


TABLE II. TOP LEVEL CATEGORIES, LABELS AND THEIR SUPPORT

LABEL	CATEGORY	SUPPORT
1	Beauty	1111099
2	Jewelry	2740585
3	Tools & Home Improvement	1816066
4	Clothing & Accessories	5479922
5	Electronics	3158641
6	Home & Kitchen	2734394
7	Grocery & Gourmet Food	371158
8	Baby Products	152314
9	Sports & Outdoors	1553128
10	Shoes	1088813
11	Office Products	900799
12	Arts Crafts & Sewing	430080
13	Patio Lawn & Garden	505820
14	Health & Personal Care	832663
15	Automotive	2518833
16	Musical Instruments	112292
17	Industrial & Scientific	481073
18	Toys & Games	536563
19	Cell Phones & Accessories	771096
20	Video Games	15631
21	Clothing	33515
22	Watches	83591
23	Software	24410
24	Pet Supplies	270278
25	Baby	2219
26	Appliances	31059
27	Home Improvement	5677
28	Kitchen & Dining	3043

To measure the amount of performance improvement achieved by the Hierarchical Classification scheme over flat single classifier approaches, a Single Level Hierarchical Classifier(SLHC) was trained.

Fig. 2. Modifying Taxonomy for SLHC



In this method, the top level classifier was retained, but all the subsequent levels, except the leaf level, were skipped. For each of the top level categories, a flat classifier was trained on the leaf level data, that fall under that parent category. This is equivalent to modifying the taxonomy by retaining the top level nodes, collapsing all other levels, except the leaf levels and attaching the leaf level nodes to their respective level one nodes, as shown in Fig 2.

As shown in Table III, the performance of single level hierarchical classification is not as high as that of complete hierarchical classification.

TABLE III. BASELINE PERFORMANCE MEASURES

Scheme	No. of Classifiers	Average test f_1 - score
SLHC	28	0.696

VI. FUTURE WORK

The remaining levels need to be trained for Amazon data. In addition, for the minority categories which were skipped, we need to use some schema-based approach. Also, we need to collect product data from other taxonomies like Walmart, eBay, Target etc., and train hierarchical classifiers on them. Once these are done, we can consider each of these taxonomies as a reference taxonomy, and implement the voting-based alignment algorithm and obtain a one-to-one mapping between categories from different taxonomies. This will also handle the case, where a single reference taxonomy containing every possible product category, might not be available.

REFERENCES

[1] K. Selcuk Candan, Mario Cataldi, Maria Luisa Sapino, and Claudio Schifanella, “Structure- and Extension-Informed Taxonomy Alignment”, VLDB Workshop on Ontologies-based Techniques for Databases in Information Systems and Knowledge Systems (ODBIS), pp. 1-8, 2008.

[2] S. T. Dumais and H. Chen, “Hierarchical Classification of Web Content”, Proceedings of SIGIR’00, August 2000, pp. 256-263.

[3] Philip Resnik, “Using Information Content to Evaluate Semantic Similarity in a Taxonomy”, Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI), 1995.

[4] Do, H. H., Rahm, E. (2002). “COMA - A system for flexible combination of schema matching approaches, Proc. 28th Int. Conference on Very Large databases (VLDB), Hongkong, Aug. 2002.

[5] Madhavan, J., P.A. Bernstein, E. Rahm, “Generic Schema Matching with Cupid”, VLDB 2001

[6] Li, W., C. Clifton, “SemInt: A Tool for Identifying Attribute Correspondences in Heterogeneous Databases Using Neural Network”, Data and Knowledge Engineering 33: 1, 49-84, 2000

[7] Babbar R., Partalas I., Gaussier ., Amini M.-R, “On Flat versus Hierarchical Classification in Large-Scale Taxonomies”, 27th Annual Conference on Neural Information Processing Systems (NIPS 2013)

[8] R.Miller, L.Haas, M.A.Hernandez, ”Schema Mapping as Query Discovery”,VLDB 2000, pp.77-88.

[9] R.E. Fan, K.W. Chang, C.J. Hsieh, X.R. Wang, and C.J. Lin, ”LIBLINEAR: A Library for Large Linear Classification”, Journal of Machine Learning Research 9(2008), 1871-1874.