Exploiting Microdata Annotations to Consistently Categorize Product Offers at Web Scale

Robert Meusel, Anna Primpeli, Christian Meilicke,
Heiko Paulheim, and Christian Bizer

Data and Web Science Group, University of Mannheim, Germany
{robert,anna,christian,heiko,chris}@dwslab.de
http://dws.informatik.uni-mannheim.de

Abstract. Semantically annotated data, using markup languages like RDFa and Microdata, has become more and more publicly available in the Web, especially in the area of e-commerce. Thus, a large amount of structured product descriptions are freely available and can be used for various applications, such as product search or recommendation. However, little efforts have been made to analyze the *categories* of the available product descriptions. Although some products have an explicit category assigned, the categorization schemes vary a lot, as the products originate from thousands of different sites. This heterogeneity makes the use of supervised methods, which have been proposed by most previous works, hard to apply. Therefore, in this paper, we explain how *distantly supervised* approaches can be used to exploit the heterogeneous category information in order to map the products to set of target categories from an existing product catalogue. Our results show that, even though this task is by far not trivial, we can reach almost 56% accuracy for classifying products into 37 categories.

Key words: Microdata, RDFa, Structured Web Data, Classification

1 Introduction

Over the last years, more and more websites started making use of markup languages like RDFa, Microformats and Microdata to semantically annotate entities describing for example events, products, organizations, and persons within their HTML pages. Those annotations can be parsed, and as they make use of well-defined vocabularies also interpreted by machines. With the amount of (freely) available data, this data space becomes more and more interesting in order to create new knowledge bases, enrich existing knowledge bases, or use the data to improve applications.

In particular, as recent studies have shown, especially such annotations have become more and more common in the e-commerce domain.[1] An important step

[1] http://webdatacommons.org/structureddata/index.html#toc3

towards exploiting that data is to obtain a detailed profiling of the data that is available. Besides basic information as the number of different instances or the amount of missing information, the category of the products and the distribution of the categories in a dataset is very important.

Although the most common vocabulary used for semantic annotations in the Web, *schema.org*, allows the markup of category information for a product description, the sites across the Web do not make use of one global homogeneous categorization schema. Instead, most sites use their own categorization systems. In order to get a unified view on the categories of products, most previous works made use of supervised methods. Those methods depend on data which is already annotate with the categories from a given target schema. Since a unified set of categories is necessary (and desired), the categories which are annotated cannot be exploit directly for this type of method, and an additional-manual annotation is necessary. As this work is most of the time costly, we propose to use distant supervision as an alternative, using the existing category systems as input. The potential advantages are the reduction of manual work in creating labeled data and keeping the data up-to-date.

The rest of this paper is structured as follows: First, we give a brief introduction to the deployment of (product-related) markup languages and vocabularies in the Web. The next section introduces the corpus we use to evaluate the supervised and distant supervised approaches which can be used to categorize the products. In the following section we state the results for a supervised approach, and then, we explain how our proposed approach, which is based on the idea of distant supervision, can be used to omit the necessity of manually labeled data to train a predictive model. After discussing related work in Section 5, we explain the benefits and drawbacks of our proposed method and line out further open challenges in the conclusion.

2 Statistics on Deployed Product-related Schema.org Microdata in the Web

In this section, we give a brief overview of the available marked up data within the Web. As an object of this analysis, we use the latest extraction of WebDataCommons (WDC), which includes over 5 billion marked up entities by one of the three main markup languages and has been retrieved from the CommonCrawl corpora of December 2014.[2] From these data, Table 1 describes the four major vocabularies which are used to describe product-related information, together with the specific classes and the number of deploying pay-level domains (PLDs).[3]

[2] `http://blog.commoncrawl.org/2015/01/december-2014-crawl-archive-available/`

[3] Similar to our previous works [9], we will analysis the data based on PLDs embedding certain vocabularies, classes and properties.

Table 1. Most common product-related deployed classes and vocabularies by number of PLDs in the 2014 corpus.

Major Markup Format	Vocabulary	Related Classes	# of PLDs
Microdata	schema.org	Product, Offer	98 608
Microdata	data-vocabulary.org	Product, Offer	16 003
RDFa	Open Graph Protocol	product	14 592
RDFa	purl.org/goodrelations	Offering	2 196

Good Relations is the original ontology which was later adopted by *schema.org* (`s:Offer` and `s:Product`) to model product-related classes.[4] PLDs (e.g. `swimoutlet.com`, `surveillance-video.com`, and `craftsman.com`) annotating information using the class `gr:Offering` make use of the property `gr:name` in 37% of the cases, but in 70% of the cases, they the property `gr:description`. This might be an effect of the sometimes rather small number of crawled pages of non-popular sites.

Open Graph Protocol is mainly used by Facebook to integrate external entities into the Facebook eco-system. From the sites making use of the `product` class in this vocabulary (e.g. `bebe.com` or `epicsports.com`), over 70% also mark the `title`, `image`, `url`, and `description`.

Data-vocabulary.org is used within Microdata. It is the predecessor of *schema.org* and is still adopted widely in the Web. Among more than 9 000 PLDs still using the `dv:Product` class in our corpus, we find also well known domains like `samsung.com` and `audible.com`. Similar to `gr:Offering`, only a small fraction ($< 50\%$) of the PLDs make use of `dv:name`, and only one third annotate a `dv:description`.

Schema.org is the most frequently used vocabulary to describe products. 89 608 PLDs (10.9%) annotate at least one entity as `s:Product` and 62 849 PLDs (7.6%) annotate at least one entity as `s:Offer`.

Table 2 shows the number and percentage of the most common used and three selected properties embedded by PLDs making use of the classes `s:Product` and `s:Offer`. Especially for the focus of the paper, the PLDs making use of `s:category` and `s:breadcrumb` are important. Here, we see only a small number of PLDs at all which annotate this information.

From the PLDs making use of *schema.org*, we identified 43 frequently visited e-commerce sites based on the reports by *Bloomberg*[5], and *Yahoo! Finance*[6] and the traffic volume of those PLDs (based on Alexa[7]). From those identified PLDs, `shopping.yahoo.com`, `hm.com`, and `oodle.com` were not contained in the original crawl (e.g. due to restrictions within the *robots.txt*). Eight of the

[4] `http://blog.schema.org/2012/11/good-relations-and-schemaorg.html`
[5] `http://www.bloomberg.com/ss/08/11/1107_ecommerce/12.htm`
[6] `http://finance.yahoo.com/news/research-markets-worlds-leading-e-154500570.html`
[7] `http://www.alexa.com/`

Table 2. Most common used and selected properties by PLDs deploying `s:Product` or `s:Offer`. * marks properties defined for `s:Product`. ** marks properties defined for `s:Offer`. *** marks properties defined for `s:WebPage`

Property	# Product PLDs	% Product PLDs	# Offer PLDs	% Offer PLDs
`s:name`*	78 292	87.37%	54 193	86.23%
`s:image`*	59 445	66.34%	45 824	72.91%
`s:description`*	58 228	64.98%	42 730	67.99%
`s:offers`*	57 633	64.32%	55 630	88.51%
`s:price`**	54 290	60.59%	59 452	94.59%
`s:availability`**	36 789	41.06%	37 871	60.26%
`s:priceCurrency`**	30 610	34.16%	32 114	51.10%
`s:url`*	23 723	26.47%	15 601	24.82%
`s:aggregateRating`*	21 166	23.62%	12 325	19.61%
`s:category`**	1 479	1.65%	1 667	2.65%
`s:breadcrumb`***	431	0.48%	460	0.73%

remaining 40 embed Microdata, but do not use product-related classes, for example `amazon.com`, which annotates the videos of their instant view platform, but not the physical products. We divided the remaining 32 into the three e-commerce roles: *producer*, *merchant* and *marketplace*, and analyzed the usage of the most common properties (based on all sites making use of products-related markup).

The results of this analysis can be found in Table 3. We found that except of the usage of `s:description` by the identified marketplaces, it is more likely that the identified e-commerce pages make use of the selected properties to annotate their products, than all product-related sites in general. In comparison to each other, merchants use the selected properties slightly more often.

Table 3. Analysis of property usage of selected 32 e-commerce sites.

	Producer	Merchant	Marketplace	Overall
`s:name`	87.5%	93.8%	87.5%	87.4%
`s:image`	75.0%	75.0%	62.5%	66.3%
`s:description`	75.0%	56.3%	37.5%	65.0%
`s:offers`	75.0%	81.3%	62.5%	64.3%
`s:price`	75.0%	75.0%	62.5%	60.6%
`s:priceCurrency`	25.0%	56.3%	50.0%	34.2%
`s:availability`	25.0%	56.3%	50.0%	41.1%

Based on these findings, such more frequently visited sites are a good entry point to gather product descriptions with a minimal set of properties.

In the following we explain how the annotated data can be used to assign categories for a given set of products. Therefore, we first introduce the data we use for evaluation and further explain our proposed approach step by step.

3 Experimental Setup

In this section, we describe the data and categorization schema used in our experiments, as well as the gold standard and the evaluation measures used.

3.1 Product *Schema.org* Microdata

From the whole WDC 2014 Microdata corpus[8], we derived a subset of 9 414 product descriptions from 818 different PLDs. We have chosen products from PLDs for which each product description uses at least the properties `s:name`, `s:description`, and `s:brand`, and either one of the two properties `s:category` (84% of the PLDs) or `s:breadcrumb` (16% of the PLDs). From each PLD, we extracted at most 20 products to reduce the risk of a bias towards a certain category. Table 4 shows an excerpt of the data. Especially for the categories/breadcrumb values, we observed a mixture of multi-level and flat paths, as well as tag-like annotations. 3 653 respectively 1 019 distinct `s:category` values respectively `s:breadcrumb` values are used by the included products.

Table 4. Product data examples

s:name	s:decription	s:brand	s:category/s:breadcr.
ColorBox Stamp Mini Tattoo	ColorBox Stamps are easy to use and perfect for papercraft fun. [..] Not for use by children 12 years andyounger.	ColorBox	Stamps >Rubber Stamp
Cowhide Link Belt	ITEM: 9108 Your search is overfor a great casual belt for jeans or khakis. [..]	-	Accessories
Fiesta SE	Automatic, Sedan, I4 1.60L , Gas, RedVIN: 3FADP4BJ8DM1679	Ford	cars
Alabama Crimson Tide Blackout Pullover Hoodie - Black	No amount of chilly weather can keep you from supporting your team.[..]	-	Alabama Crimson Tide >40*to*60
231117-B21 HP PIII P1266 1.26GHz ML330 G2	Description:Pentium III P1266 ML330 G2/ML350 G2/ML370G2 (1.26GHz/ 133MHz/512K/43W) [..] # 231117-B21	HP Compaq	G2 Xeon
TFS Lil' Giant Anvil, 65 lb	Dimensions: Face 4" x 10.75" Horn 4" x 8.25" Height8" Base 9.25" x 11" Hardie Hole: 1" [..] #: TFS7LG65	Anvils [..]	Hardware >Tools >Anvils
Gavin Road Cycling Shoe	For great performance at adiscounted price, [..]	-	Root RoadBikeOutlet.com >Apparel >Shoes >>

3.2 GS1 - Global Product Catalogue

For our experiments, we used the *GS1 Product Catalogue* (GPC) as target hierarchy. The GPC is available in different languages and claims to be a standard for everything related to products.[9] The hierarchy is structured in six different levels starting from the *Segment*, over *Family*, *Class*, and *Brick*, down to the last

[8] `http://webdatacommons.org/structureddata/2014-12/stats/schema_org_subsets.html`

[9] `http://www.gs1.org/gpc`

two levels *Core Attribute Type* and *Core Attribute Value*. The first level distinguishes between 38 different categories, the second level divides the hierarchy into further 113 categories and the third level consists of 783 disjunct categories. In addition to the textual labels for each category in the hierarchy, the forth and the sixth level partly include a – more or less – comprehensive textual description. Table 5 shows the first four levels of three paths of the hierarchy.

Table 5. Excerpt of GS1 GPC (first four levels). [..] is a placeholder, if the label is similar to the one of the former level.

Segment	Family	Class	Brick
Toys/Games	[..]	Board Games/Cards/Puzzles	Board Games (Non Powered)
Food/Beverage/Tobacco	Seafood	Fish Prepared/Processed	[..] (Perishable)
Footwear	[..]	Footwear Accessories	Shoe Cleaning

3.3 Gold Standard

Using the set of categories from the previously mentioned hierarchy, we manually annotated the set of products described in Section 3.1. Specifically, we annotated each product (if possible) with one category for each of the first three levels. The annotations were performed by two independent individuals. Whenever there was a conflict, a third person was asked to solve the discrepancy. The annotators were first asked to read the title/name, description, and the additional available values of the product and, in case of insufficient information, they should visit the web page of the product.

Within the gold standard, we could not assign any category to 187 (2.09%) products, mostly because the available attributes to describe the products were insufficient, and the web page of the product was either not available any more or here also not enough information were given. Based on the first level of the GS1 GPC hierarchy, we assigned at least each category once (except *Cross Segment*). Table 6 depicts the ten most frequent categories of the first level within the gold standard. We see a domination by the category *Clothing*. For the second level, we assigned 77 (68.14%) different labels at least once, and 303 (38.70%) different labels for the third level. The gold standard, as well as more comprehensive statistics, can be found at our website.[10]

3.4 Baseline and Evaluation

As we want to show to which extent the categorizations of the single PLDs can be used to assign categories from a global hierarchy to products, we compare ourselves to the results of a supervised classification approach. The approach is trained using 10-fold cross-validation with three different classification methods: Naive Bayes (NB), Decision Trees (DT) and k-Nearest Neighbor approach, where

[10] `http://webdatacommons.org/structureddata/2014-12/products/gs.html`

Table 6. Distribution of categories for the first Level of the GS1 GPS within the gold standard dataset, as well as with the predicted distributions of the best supervised and distant supervised approach.

Rank	Category Level 1	Original	Supervised	Δ	Distant Superv.	Δ
1	Clothing	0.435	0.401	0.033	0.406	0.028
2	Personal Accessories	0.053	0.128	0.075	0.039	0.014
3	Household/Office Furniture/Furnishings	0.051	0.045	0.006	0.035	0.016
4	Automotive	0.047	0.054	0.007	0.052	0.005
5	Computing	0.037	0.034	0.004	0.023	0.014
6	Audio Visual/Photography	0.036	0.030	0.006	0.020	0.015
7	Healthcare	0.033	0.027	0.006	0.005	0.027
8	Pet Care/Food	0.026	0.028	0.002	0.017	0.010
9	Sports Equipment	0.026	0.030	0.004	0.022	0.004
10	Food/Beverage/Tobacco	0.024	0.025	0.001	0.007	0.018
11-38	*Others*	0.232	0.198	0.065	0.373	0.159

$k = 5$ (5-NN). A detailed description of the baseline method can be found in Section 4.2.

For reasons of comparison, we use *accuracy* (ACC) as the main evaluation metric. Whenever an approach is not able to return a category for a given product, we count this examples as a *false negative*. For approaches returning either one label or no label for each instance, this measure is equal to recall (R). In addition, for our distant-supervised approaches, we also report the precision P, as this measure gives an idea about the performance of predicted labels, without regard of those which cannot be labeled. We also state the f-score $F1$, representing a trade-off between R and P.

4 Experiments & Results

In this section, we will first state how the both input data sources, i.e., product descriptions and categories from a given target hierarchy, are transformed into feature vectors that can be processed by further methods. Then, we train a model based on the hand-annotated categories of the gold standard. The remaining parts of this section introduce our distant supervision approach making use of the categorical information for the products given on the PLDs itself.

4.1 Feature Vector Generation

As stated above, we have two types of input: products, which are described by a set of properties, and the categories of the target hierarchy. In order to transform both types of input into comparable feature vectors, we generate a *bag of words* representation for each entity, i.e., each product and each category at a certain depth within the hierarchy.

For the products, we experiment with different sets of property combinations (e.g. only `s:title`, `s:title` with `s:description`, and so on). For the hierarchies, we use the textual names of the categories themselves and all or a selection of the names of sub-categories (e.g., segment, segment and family, segment and

brick). In all cases, we tokenize the textual values by non alpha-numeric characters, remove stopwords and stem the data using a *Porter Stemmer*. Moreover, we transform all characters to lower case and remove terms which are shorter than 3 and longer than 25 characters.

In order to weight the different features for each of the elements in the two input sets, we apply two different strategies:

Binary Term Occurrence (BTO), where the weight of a term for an element is either 1, if the term occurs at least once within the textual attributes of the element, 0 otherwise.

TF-IDF, where the term frequency is normalized by the inverse document frequency, which removes the impact of common terms which occur in a large fraction of documents.

In the following, we refer to the set of feature vectors describing products by *Pro* and to those describing labels of the categories of the hierarchy by *Cat*. Depending on the textual attributes which were used to create the vectors, the number of final attributes ranges between 4 000 (only category and breadcrumb) to around 11 000 (all properties).

4.2 Baseline: Supervised Approach

Table 7 presents the results with different setups for the baseline classification approach. We reach the highest accuracy with a 5-NN classification algorithm using *Jaccard Coefficent*. Decision Trees did not perform at a comparable level, so we excluded them from the table. We also calculated the distribution of the predicted product categories for the best approach. The results are shown in Table 6 including the deviation from the distribution of categories in the gold

Table 7. Selected results of the baseline classification for assigning GS1 GPC first level categories. Highest scores are marked in **bold**.

Selected Properties	Term Weight.	Classifier	ACC
Name,Desc	BTO	NB	.722
Name,Description	TF-IDF	NB	.733
Name,Description	BTO	5-NN(Jaccard)	.608
Name,Description	TF-IDF	5-NN(Cosine)	.728
Name,Description,Categroy,Breadcr.	BTO	NB	.754
Name,Description,Categroy,Breadcr.	TF-IDF	NB	.757
Name,Description,Categroy,Breadcr.	BTO	5-NN(Jaccard)	.819
Name,Description,Categroy,Breadcr.	TF-IDF	5-NN(Cosine)	.740
Name,Description,Categroy,Breadcr.,Brand	BTO	NB	.758
Name,Description,Categroy,Breadcr.,Brand	TF-IDF	NB	.760
Name,Description,Categroy,Breadcr.,Brand	BTO	5-NN(Jaccard)	**.820**
Name,Description,Categroy,Breadcr.,Brand	TF-IDF	5-NN(Cosine)	.746

4.3 Hierarchy-Based Product Classification

In a first step, we use the feature vectors created for the categories from the target hierarchy *Cat* in order to train a predictive model (one labelled example

for each category). This model is then used to predict the labels for the instances of *Pro*. We test different classification methods, namely *Naive Bayes* (NB), *k-Nearest-Neighbour* with $k = 1$ (1−NN)[11], *Support Vector Machines* (SVM), and *Random Forests* (RF).

Table 8 shows the results of the best configuration, using only the features from the values of the properties name, category and breadcrumb from *Pro* and all hierarchy labels from the GS1 GPC. We find that on average TF-IDF as term weighting methods performs better than a BTO strategy. The best results are achieved using 1-NN and Naive Bayes classification on TF-IDF vectors.

Table 8. Best results achieved with distant supervised classification using instances of *Cat* for training. Highest scores are marked in **bold**.

Term Weighting	Classifier	ACC
TF-IDF	NB	**.377**
TF-IDF	1-NN (Cosine)	**.377**
TF-IDF	1-NN (Jaccard)	.361
TF-IDF	SVM	.376
TF-IDF	Random Forest	.006
BTO	NB	.000
BTO	1-NN (Cosine)	.330
BTO	1-NN (Jaccard)	.271
BTO	SVM	.000
BTO	Random Forest	.026

4.4 Similarity-based Product Category Matching

In order to exploit the promising performance of the distance-based classification approach (1-NN) of the former section, we extend our approach in this direction, using the similar fundamental idea as nearest-neighbour classifier. We calculate for each instance in *Pro* the distance to all instances in *Cat*. To that end, we use three different similarity functions, namely:

Cosine Similarity: This measure sums up the product of the weights/values for each attribute of the two vectors and is supposed to work well with TF-IDF.

Jaccard Coefficient: This measure calculates the overlap of terms occurring in both vectors and normalize it by the union of terms occurring in both vectors. This measure is supposed to work well with binary weights.

Non-normalized Jaccard Coefficient: As the descriptions of products could be rather comprehensive (based on the way the data was annotated), we address the penalization of longer product names, which would occur for Jaccard, by introducing a non-normalized version of the *Jaccard-Coefficent*, i.e., only measuring the overlap.

[11] As for each class, only one example exists k needs to be set to 1, otherwise the method would consider other examples then the nearest, which by design belong to another class. This setup is equal to *Nearest Centroid Classification*, where each feature vector of *Cat* is equal to one centroid.

In addition, we use different sets of textual attributes from the products as well as from the hierarchies to build the feature vectors. Based on the similarity matrix, we then select for each instance in Pro the instance in Cat with the highest score, larger than 0. In contrast to a classifier, we do not assume any distribution within the data, or assign any category randomly. This means in case of two or more possible categories which could be assigned, we do not assign a particular instance from Cat to the instance of Pro.[12]

Table 9 reports a selection of results of this approach trying to predict the categories of the first, second and third level within the hierarchy. In each of the three blocks, the first line always reports the best results using only the category and breadcrumb as input for the feature vector. The second line reports the result for the default configuration (all attributes, TF-IDF). The third line shows the result for the optimized setup of attributes and term weighting. In all cases in the table cosine similarity produced the best results. In some experiments the other two similarity functions performed comparable, but overall did not produce better results. Starting from level one to three, we see a slight decrease in terms of accuracy. This is not surprising as the number of possible labels increases with each level (see Section 3.3) and the contentual boundaries between them become more and more fuzzy. In addition, we found that the best configuration just differs by some percentage points from the default configuration for all three levels (e.g. .341 vs. .359 for the first level). Furthermore, using only the information from the category and the breadcrumb alone does not produce the highest accuracy results. For all three levels the best results in terms of accuracy could be reached using the textual values of category, breadcrumb and name as input for the feature vector creation.

Table 9. Selected results for all three category levels, including the default configuration, the best with and without ground knowledge.

Level	Product Properties	Product Weight.	Hierarchy Term Levels	Hierarchy Term Weight.	Ground Knowledge	ACC	P	F1
1	Category, Breadcr.	BTO	1-6	TF-IDF	none	0.288	0.334	0.309
1	All	TF-IDF	1-6	TF-IDF	none	0.341	0.344	0.343
1	Name, Category, Breadcr.	BTO	1-6	TF-IDF	none	0.359	0.373	0.366
1	Name, Category, Breadcr.	BTO	1-4	TF-IDF	DISCO	0.479	0.499	0.489
2	Category, Breadcr.	BTO	1-6	TF-IDF	none	0.171	0.297	0.217
2	All	TF-IDF	1-6	TF-IDF	none	0.261	0.264	0.263
2	Name, Category, Breadcr.	BTO	1-6	TF-IDF	none	0.294	0.305	0.300
2	Name, Category, Breadcr.	BTO	1-4	TF-IDF	DISCO	0.380	0.395	0.387
3	Category, Breadcr.	BTO	1-6	TF-IDF	none	0.109	0.112	0.111
3	All	TF-IDF	1-6	TF-IDF	none	0.196	0.198	0.197
3	Name, Category, Breadcr.	BTO	1-6	TF-IDF	none	0.257	0.267	0.262
3	Name, Category, Breadcr.	BTO	1-4	TF-IDF	DISCO	0.258	0.269	0.263

Inspecting the results of the optimal solution for each level manually, we found that in most cases the overlap in features between the instances of Pro and Cat was insufficient for those instances which were wrongly categorized or

[12] As stated before, such instances are counted as *false negatives* within the evaluation.

left unlabelled. Reasons for this are the use of a different language as the target hierarchy (e.g. Spanish), a different granularity (e.g. *fruits* versus *cherry*) or the use of synonyms (e.g. *hat* versus *cap*).

A common method to overcome at least the two latter discrepancies is the enhancement with a external/additional ground knowledge.[13] For our experiments, we use two different sources of ground knowledge to enhance our feature vectors. First, we make use of the *Google Product Catalogue*.[14] This catalog is used by Google to enable advertisers to easily categorize the ads they want to place. The catalog is available in different languages and in addition is more tailored towards products traded in the Web. The second source we use is based on the co-occurrences of different terms within a corpus. In particular, we make use of *extracting DIStributionally related words using CO-occurrences* framework (DISCO)[15], first presented by Kolb [5], where we load the English Wikipedia and enhance the feature vectors of the categories.

The best results and the comparison to the best results without the enhancement can also be seen in Table 9, within the third and forth row of each block. In general we find a strong increase in the accuracy in comparison to the non-enhanced experiments. For the first level, we increase our performance by 33% to almost .5 accuracy. For level three, however, this effect diminishes almost completely. Even with the enhanced vectors, the improvements are small.

In the following we describe two different types of experiments to further improve our results. In the first, we concentrate on high-precision results and obtained those values as labeled instances. Then, we train a predictive model on those instances. In the second approach, we reformulate the task of labeling a set of instances as a global optimization problem.

4.5 Classification on High-Precision Mappings

This approach is based on the idea that, even if the accuracy (which represents the global performance of the matching) is not sufficient, we could make use of those instances which were assigned to a category with a high precision. Those instances can further be used as input in order to train a predictive model. It is important to note that when selecting the mapping, all, or at least a large fraction of categories (which should be predicted), should be included. This means that some configurations even with $P = 1$ are not useful, as they include too few instances. In order to improve the precision of our initial mapping, we introduce a higher minimal similarity between products and categories.

The first columns in Table 10 show the highest precisions which could be reached, where at least 100 product instances were assigned to an instance of Cat of level 1. The precision of those optimal configurations ranges between .75 and .79, which means that within this data, every fourth or fifth instance is

[13] We thank Stefano Faralli for his valuable feedback and recommendations.

[14] `https://support.google.com/merchants/answer/1705911?hl=en`

[15] `http://www.linguatools.de/disco/disco_en.html`

wrongly labeled. In addition, we report the values for a setup with less precision (.57) but with over 5 500 labeled examples.

We tested different mappings and train different classification methods on this input data. In Table 10 we outline the best performing results for the different input configurations.[16]

Table 10. Result of combined approach, using high-precision mapping result as training data to learn a predictive model for level 1 categorization.

Product	Product	Hierarchy	Term	Ground	Min.	Mapping				Overall
Properties	Weight.	Levels	Weight.	Knowldg.	Sim.	ACC	P	# Inst.	Classifier	ACC
Name,Cat.,Breadcr.	BTO	1-6	TF-IDF	Google	>.35	0.009	0.789	109	NB	0.076
All	BTO	1-6	TF-IDF	Google	>.25	0.008	0.772	103	5-NN	0.079
Name,Cat.,Breadcr.	TF-IDF	1-6	TF-IDF	Google	>.25	0.028	0.747	340	NB	0.069
Name,Cat.,Breadcr.	BTO	1-4	TF-IDF	Disco	>.05	0.340	0.570	5 505	RF	0.514

We found, that in case of the high-precision configurations (first three rows) the overall precision of the classifier which can be trained based on those input data is poor, and in all three cases did not exceed 10% accuracy. Manually inspecting those datasets and the resulting classifications reveals that not all classes are included in those sets, so the model cannot predict all classes (as they are unknown) and that the number of training data is not enough even for the classes which are included. Inspecting the results of the fourth configuration, where the final accuracy exceeds slightly the 50%, we found almost a balanced distribution in the errors of the classification.

4.6 Global Optimization

In the approaches so far, we evaluate each match between an instance in *Pro* and *Cat* in isolation. However, the similarity between two products should be used as an indicator for mapping these instances to the same category, and vice versa. Deciding about the similarity of products and matching them to categories are thus highly dependent problems.

We try to take these dependencies into account by formalizing the problem as a global optimization problem in terms of Markov Logic [2]. In particular, we use the solver *RockIt* by Noessner et al. [13] to compute the most probable solution, also known as the MAP (maximum a posteriori) state. In our formalization, we use two weighted predicates *map* and *sim* to model the mapping of a product to a category and to model the similarity of two products. We use the similarity matrices from the former experiments as input for defining the prior weights for the respective atoms. Then we define the score which needs to be optimized as the sum the weights attached to these atoms. Further, we define two hard constraints which have to be respected by any valid solution. (1) If two products are similar they have to mapped to the same category. (2) Each product can be assigned to only one category.

[16] We also applied up-sampling of under-represented classes in the dataset, but the results did not improve.

Using the best configuration of the former similarity-based matching results from Section 4.4, where we reached an accuracy level of .479, we tested different combinations for the similarity of products, as well as the minimal similarity we handed into the global optimization problem. In addition, we also tested different weight ratios between the two predicates, where we multiply the original weight of *map* with a constant factor. In Table 11, we report the best configurations and the corresponding accuracy values. In comparison to the original value of .479 we could improve up to .555, and we assume that this is not the best value which can be reached. Unfortunately, even if running the solver on a large machine, with 24 cores and over $300GB$ of RAM, further experiments did not finish within 24 hours, which shows that the approach is promising, but requires more tweaks to run at large scale.

We have selected the best performing distant supervised approach and calculated again the resulting distribution of product categories (see Table 6). Note that the supervised approach has a summed absolute error of .20 while the best distant supervised approach has a summed absolute error of .31 (the average absolute error is .006 respective .008).

Table 11. Results of the best configurations for solving the optimization problem. Highest scores are marked in **bold**.

similarity		min. value	weight ratio			
map	sim	for sim	map/sim	ACC	P	$F1$
Cosine	Cosine	0.5	20/1	0.505	0.540	0.522
Cosine	Jaccard	0.5	20/1	0.483	0.506	0.494
Cosine	Cosine	0.5	10/1	0.514	0.556	0.534
Cosine	Jaccard	0.5	10/1	0.484	0.509	0.496
Cosine	Cosine	0.4	10/1	0.553	0.606	0.578
Cosine	Cosine	0.3	10/1	**0.555**	0.636	0.593

5 Related Work

In this section, we describe relevant works both in the area of analyzing the deployment of structured web data in general, as well as in automatic product classification.

5.1 Deployment of Structured Data

The deployment of structured data was first presented by Mika and Potter [10, 11] and later an updated view on the adoption of schema.org was given by Guha [4], where [14] analyzed this vocabulary on schema level. In our previous works we analyzed the deployment of the three markup languages in a general matter [1, 9], and in addition analyzed the kinds of errors included in such kind of data [8]. Besides, we also inspected how the deployment changes over time [7].

In addition to those markup languages, recent works try to leverage information embedded in HTML tables [3, 6, 17].

5.2 Categorization of Product Data

Learning a classification model to predict labels for unclassified products was presented in [15]. The authors made use of products and their categories retrieved from `amazon.com`. Our proposed approaches aims at removing the dependency on external data classification providers.

A recent approach by Qiu et al. [16] presented a system which efficiently detects product specifications from product detail pages for a given category. In order to determine the category, they make use of pre-trained classifiers and a set of seed product identifiers of products related to this category.

Nguyen et al. [12] present an end-to-end solution facing the problem of keeping an existing product-catalogue with several categories and sub-categories up-to-date. Their approach includes data extraction, schema reconciliation, and data fusion. The authors show that they can automatically generate a large set of product specifications and identify new products.

The three mentioned approaches make use of hand-labeled or pre-annotated data, which is not (easily) accessible in larger quantities. This underlines the need of alternative methods to overcome the need of labeled data.

6 Conclusion

In this paper, we have first given a short overview about the deployment of product-related markup languages and vocabularies within HTML pages. In the second part, we have described a subset of this data, which we manually annotated with the categories of the first three levels of the GS1 Global Product Catalogue. Based on that gold standard, we have shown that using supervised methods can reach an accuracy of 80% when learning a predictive model in order to categorize products.

Further, as already some sites mark products with a site-specific category, we first have shown that using this information alone, due to its heterogeneity among different sites, is not an optimal input for a distantly supervised approach. But in combination with other properties (e.g. the name), that information can be leveraged by distantly supervised methods and thereby assign categories from a given set to products with an accuracy of up to 56%. To that end, we use various refinements of the problem, taking both background knowledge into account, as well as modeling the categorization of a set of instances as a global optimization problem. The latter provides very promising results, but also hints at scalability issues of solving such optimization problems.

Regarding the distributions which are predicted by the two different kinds of approaches, we see that the supervision works slightly better, but both results can be used in order to gain first insights in the category distribution of the dataset.

Another area where further improvements can be made is the selection of sources. In our gold standard, we only included product descriptions from less than 1 000 PLDs, while on the Web, there are by far more which can be exploited.

In particular it might be a promising approach to weight the influence of products of a particular PLD by other attributes, for example the average length of the description or the depth of the given category information.

References

1. C. Bizer, K. Eckert, R. Meusel, H. Mühleisen, M. Schuhmacher, and J. Völker. Deployment of rdfa, microdata, and microformats on the web – a quantitative analysis. In *ISWC*. Springer, 2013.
2. P. Domingos and D. Lowd. Markov logic: An interface layer for artificial intelligence. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 3(1):1–155, 2009.
3. J. Eberius, M. Thiele, K. Braunschweig, and W. Lehner. Top-k entity augmentation using consistent set covering. SSDBM '15, 2015.
4. R. V. Guha. Schema.org update. `http://events.linkeddata.org/ldow2014/slides/ldow2014_keynote_guha_schema_org.pdf`, April 2014.
5. P. Kolb. Disco: A multilingual database of distributionally similar words. In *In Proceedings of KONVENS*, 2008.
6. O. Lehmberg, D. Ritze, P. Ristoski, R. Meusel, H. Paulheim, and C. Bizer. The mannheim search join engine. *Web Semantics: Science, Services and Agents on the World Wide Web*, 2015.
7. R. Meusel, C. Bizer, and H. Paulheim. A web-scale study of the adoption and evolution of the schema.org vocabulary over time. In *Proc. WIMS'15*, pages 15:1–15:11, New York, NY, USA, 2015. ACM.
8. R. Meusel and H. Paulheim. Heuristics for fixing errors in deployed schema.org microdata. In *Extended Semantic Web Conference*, 2015.
9. R. Meusel, P. Petrovski, and C. Bizer. The webdatacommons microdata, rdfa and microformat dataset series. In *ISWC*, 2014.
10. P. Mika. Microformats and RDFa deployment across the Web . `http://tripletalk.wordpress.com/2011/01/25/rdfa-deployment-across-the-web/`, 2011.
11. P. Mika and T. Potter. Metadata statistics for a large web corpus. In *LDOW 2012*, CEUR Workshop Proceedings, Vol. 937. CEUR-ws.org, 2012.
12. H. Nguyen, A. Fuxman, S. Paparizos, J. Freire, and R. Agrawal. Synthesizing products for online catalogs. *Proc. VLDB Endow.*, 4(7):409–418, Apr. 2011.
13. J. Noessner, M. Niepert, and H. Stuckenschmidt. Rockit: Exploiting parallelism and symmetry for MAP inference in statistical relational models. In *Proc. AAAI'13*, 2013.
14. P. F. Patel-Schneider. Analyzing Schema.org. In *International Semantic Web Conference*, 2014.
15. P. Petrovski, V. Bryl, and C. Bizer. Integrating product data from websites offering microdata markup. In *DEOS2014*, 2014.
16. D. Qiu, L. Barbosa, X. L. Dong, Y. Shen, and D. Srivastava. Dexter: Large-scale discovery and extraction of product specifications on the web. *Proceedings of the VLDB Endowment*, 8(13), 2015.
17. D. Ritze, O. Lehmberg, and C. Bizer. Matching html tables to dbpedia. In *Proceedings of the 5th International Conference on Web Intelligence, Mining and Semantics*, page 10. ACM, 2015.