The WDC Training Dataset and Gold Standard for Large-Scale Product Matching

Anna Primpeli
Data and Web Science Group
University of Mannheim
anna@informatik.uni-mannheim.de

Ralph Peeters
Data and Web Science Group
University of Mannheim
rpeeters@mail.uni-mannheim.de

Christian Bizer
Data and Web Science Group
University of Mannheim
chris@informatik.uni-mannheim.de

ABSTRACT

A current research question in the area of entity resolution (also called link discovery or duplicate detection) is whether and in which cases embeddings and deep neural network based matching methods outperform traditional symbolic matching methods. The problem with answering this question is that deep learning based matchers need large amounts of training data. The entity resolution benchmark datasets that are currently available to the public are too small to properly evaluate this new family of matching methods. The WDC Training Dataset for Large-Scale Product Matching fills this gap. The English language subset of the training dataset consists of 20 million pairs of offers referring to the same products. The offers were extracted from 43 thousand e-shops which provide schema.org annotations including some form of product ID such as a GTIN or MPN. We also created a gold standard by manually verifying 2200 pairs of offers belonging to four product categories. Using a subset of our training dataset together with this gold standard, we are able to publicly replicate the recent result of Mudgal et al. that embeddings and deep neural network based matching methods outperform traditional symbolic matching methods on less structured data.

CCS CONCEPTS

• **Information systems → Electronic commerce**.

KEYWORDS

entity resolution, product matching, schema.org annotations, deep matching, embeddings, evaluation data

ACM Reference Format:
Anna Primpeli, Ralph Peeters, and Christian Bizer. 2019. The WDC Training Dataset and Gold Standard for Large-Scale Product Matching. In *Companion Proceedings of the 2019 World Wide Web Conference (WWW'19 Companion), May 13–17, 2019, San Francisco, CA, USA.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3308560.3316609

1 INTRODUCTION

As a result of the success of using embeddings and deep neural networks for image and speech recognition as well as natural language processing, the question whether these techniques also increase the performance of entity matching methods has recently moved into the research focus [12, 17, 18]. Current results by Mudgal et al. [12] suggest that deep learning techniques perform similar to traditional symbolic matching techniques on strongly structured data but outperform traditional techniques by a margin of 5% to 10% in F1 on less structured data such as product descriptions in e-commerce. The problem with these results is that they are not publicly reproducible as they have been achieved using large training datasets from a 'major retailer' [12] which are not available to the public.

Many e-shops have started to mark-up offers in HTML pages using schema.org annotations. In recent years, many of these e-shops have also started to annotate product identifiers within their pages, such as manufacturer part numbers (MPNs), global trade item numbers (GTINs), or stock keeping units (SKUs), a practice that was less common 5 years ago. These identifiers allow offers for the same product from multiple e-shops to be grouped into clusters. The *Web Data Commons* (WDC) project[1] monitors the adoption of schema.org annotations by analysing the CommonCrawl[2], a series of public web corpora each consisting of several billion HTML pages. Table 1 shows the number of pay level domains (PLDs) in the CommonCrawl that use product-related schema.org terms in 2017 compared to 2013. We see that the absolute numbers of websites, the richness of the descriptions, as well as the number of websites annotating product identifiers (lower part of the table) have all grown significantly.

This paper presents a large public training dataset for product matching which has been produced by extracting schema.org product descriptions that include identifiers from the CommonCrawl (November 2017). The training dataset consists of 26 million offers from 79 thousand websites. Using the identifiers and a specific cleansing workflow, the offers are grouped into 16 million clusters of offers referring to the same product. 1.1 million of these clusters have a size of three and larger, 413 thousand have a size of five and larger. The English language subset of this dataset consists of 16 million offers which are grouped into 10 million clusters. Out of these clusters, 625.7 thousand have a size of three and larger and 225 thousand have a size of five and larger. Only considering clusters of English offers having a minimum size of five and excluding clusters having a size larger than 80 which may introduce noise, 20.7 million positive training examples (pairs of matching product offers) and a maximum of 2.6 trillion negative training examples can be derived from the dataset. This means that the training dataset is several orders of magnitude larger than the largest training set for product matching that has been accessible to the public so far (see Section 6).

WWW '19 Companion, May 13–17, 2019, San Francisco, CA, USA

ACM ISBN 978-1-4503-6675-5/19/05.
https://doi.org/10.1145/3308560.3316609

[1] http://www.webdatacommons.org/structureddata/
[2] http://commoncrawl.org/

Table 1: Adoption of product-related schema.org properties. The percent numbers refer to all websites using schema.org product markup.

Property	#PLDs 2013	#PLDs 2017	%PLDs 2013	%PLDs 2017
s:Product/name	50,536	535,625	89.62%	92.11%
s:Offer/price	33,509	462,444	59.42%	79.53%
s:Product/offers	33,090	462,233	58.68%	79.49%
s:Offer/priceCurrency	14,704	430,556	26.08%	74.04%
s:Product/image	34,921	419,391	61.93%	72.11%
s:Product/description	38,037	377,639	67.46%	64.94%
s:Offer/availability	21,789	337,876	38.64%	58.11%
s:Product/url	11,937	263,720	21.17%	45.35%
s:Product/brand	5,880	73,934	10.43%	12.71%
s:Product/productID	7,392	35,211	10.90%	6.05%
s:Product/sku	1,323	126,696	1.95%	21.78%
s:Product/mpn	484	8,161	0.71%	1.40%
s:Product/gtin13	276	5,467	0.41%	0.94%
s:Product/identifier	160	538	0.24%	0.09%
s:Product/gtin8	0	257	0.00%	0.04%
s:Product/gtin12	0	577	0.00%	0.09%
s:Product/gtin14	0	722	0.00%	0.12%

In addition to the training dataset, we build a gold standard for evaluating matching methods by manually verifying that 2200 pairs of offers refer or do not refer to the same products. The gold standard covers the product categories *computers, shoes, watches,* and *cameras*. Using both artefacts to publicly verify the results of Mudgal et al. [12], we find that embeddings and deep learning based methods outperform traditional symbolic matching methods by 5% to 11% in F1 on our gold standard.

This paper is structured as follows: Section 2 describes the data cleansing procedure that was used to create the training dataset. Section 3 profiles the cluster structure, the number of offers per product category, and the density of the schema.org attributes in the training dataset. We describe the creation and profile of the gold standard in Section 4. Section 5 presents the results using the datasets to train and evaluate various baseline matchers as well as the embeddings and deep learning based matchers proposed by Mudgal et al. Finally, Section 6 compares the training dataset and gold standard to existing evaluation datasets. The training set and the gold standard are provided for public download on the Web Data Commons website[3] which also provides additional statistics about both.

2 TRAINING DATASET CREATION

We use the Web Data Commons schema.org/Product data corpus November 2017[4], containing 809 million schema:Product and schema:Offer entities, as starting point for the creation of the training set. This section describes the cleansing procedure that we apply to derive the training set from the corpus. First, we focus on detecting offers having annotated identifiers and develop strategies to overcome syntactic errors in the annotations [7, 11]. Next, we group the offers using the identifiers into clusters. Finally, we extract key/value pairs from the HTML tables in the pages containing product specifications and categorize the offers into 26 product categories.

Table 2: Value overlap of different ID properties

Value overlap	gtin8	gtin12	gtin13	gtin14
sku	2,065	18,682	103,736	16,897
productID	2,966	13,854	198,847	10,192
identifier	122	3,751	17,732	837
Overlap #	5,153	36,287	320,315	27,926
Overlap %	6.75%	15.81%	11.21%	11.35%

Selection of schema.org identifier related properties. According to the schema.org/Product[5] and schema.org/Offer[6] property definitions, the terms *gtin* and *mpn* should be used to annotate global-scoped identifiers. Vendor specific identifiers should be marked-up as *sku*, while *productID* and *identifier* can be used either to markup vendor-specific or global identifiers. However, we observe that the identifier related terms are used inconsistently in many cases, as vendor-scoped terms are often used to annotate global-scoped identifier values. More specifically more than 19% of the distinct global identifier values are annotated using the property *sku* while the properties *identifier* and *productID* are also often used for the same purpose as shown in Table 2. Based on this observation, we consider all *schema.org/Product* and *schema.org/Offer* entities that include some form of identifying information using the properties *gtin8, gtin12, gtin13, gtin14, mpn, sku, identifier,* and *productID* for the creation of the training corpus.

Usage of non-existing schema.org terms. Similar to the observations of [11], we notice that 6% of the websites annotating product offers with identifier values use invalid schema.org terms. A frequent error pattern is the usage of non-existing schema.org types, such as *IndividualProduct/productID* or *ProductModel/sku.* Despite the wrong vocabulary usage such terms reveal identifying information for an offer and we do not want to ignore them upon the creation of the training set. We capture such offers by applying the following regular expression pattern on their predicates: $. * /(gtin8|gtin12|gtin13|gtin14\ |sku|mpn|identifier|productID)$. This results in 116 million offer entities being selected from the WDC schema.org/Product subcorpus.

Leveraging entity relations. Approximately 20% of the offers do not contain descriptive properties such as *schema.org/name* and *schema.org/description.* This originates in the annotation practice of describing a product and an offer for this product using two separate schema.org entities. The two entities are connected using properties such as *Product/offers* while the descriptive and identifying information is split between the two entities. We identify those entity relations and merge the descriptive information. This leads to a reduction of offers having no descriptive properties to less than 3%.

[3] http://www.webdatacommons.org/largescaleproductcorpus/

[4] http://www.webdatacommons.org/structureddata/2017-12/stats/schema_org_subsets.html

[5] https://schema.org/Product

[6] https://schema.org/Offer

Detection and removal of listing pages and advertisements. We want to include the comprehensive information about a product from its detail page into the corpus and not the summaries of this information often found on listing pages and in advertisements on other detail pages. For the detection of listing pages and advertisements a heuristic based on the following features is applied: amount of schema.org/Offer and schema.org/Product entities per webpage, variation of the length of the product descriptions, number of identifier values, and semantic connection to parent entities using the terms schema:RelatedTo and schema:SimilarTo. Our heuristic for identifying listing pages and advertisements achieves an F1 score of 94.8% on a manually annotated test set. This cleansing step removes 49% of the offer entities, leaving 58 million non listing or ad offer entities in the training set.

Filtering by identifier value length and occurrence. As next pre-processing step, the annotated identifier values are normalized by removing non-alphanumeric characters and common prefixes such as initial zero digits and identifier related strings like *ean, mpn, sku,* and *isbn.* Considering the length of global identifiers such as GTIN or ISBN numbers and the fact that short identifiers more likely introduce noise in the cluster creation phase, we filter out all offers having identifiers which are shorter than 8 characters. Additionally, offers whose id values completely consist of alphabetical characters are removed. Finally, we observe that a considerable amount of websites use the same identifier value to annotate all their offers, likely due to an error in the script generating the pages. We detect these websites and remove their offers from the training set. After these filtering steps 26 million offer entities remain in the training corpus.

Clusters creation. We group the remaining 26 million offers into 18 million clusters using their identifier values. It happens that single offers contain multiple alternative identifiers referring to the same product, e.g. GTIN8 and GTIN12, or GTIN12 and MPN. We use this information to merge clusters referring to the same product which results in a reduction of the number of clusters to 16,391,439. We also note that some websites include identifiers referring to product categories, such as UNSPSC numbers, in addition to identifiers referring to single products into the annotations. For detecting such cases, we examine the structure of the identifier co-occurrence graph within each cluster. We discover that vertices having a degree larger than 10 and a clustering coefficient of $C_i < 0.2$ tend to represent product categories rather than single products and we split the clusters accordingly. This leads to the creation of 199,139 additional clusters.

Specification tables detection and extraction. Product pages often contain specification tables describing the product in the form of key/value pairs. This structured product data is often very helpful for matching [6, 14]. Based on the work of [13, 16] on detecting specification tables in HTML pages, we apply a table detection heuristic considering the following HTML table attributes: ratio of alphanumeric characters, average number of columns per row, table children elements, image occurrences, maximum number of columns, maximum number of rows, and average length of text per row. Afterwards, we use the two column-heuristic from [16] to extract key/value pairs from tables. Evaluating our specification table detection heuristic on 455 manually annotated HTML tables, we find that the heuristic reaches an F1 of 78%.

Table 3: Distribution of offer entities and positive pairs per cluster size

Cluster size	# Clusters		# Positive Pairs	
	Full Set	English Set	Full Set	English Set
1	13,301,842	8,434,389	0	0
2	1,915,909	1,012,220	1,915,909	1,012,220
[3-4]	760,360	400,522	3,026,997	1,552,680
[5-10]	304,379	163,356	5,677,852	3,064,532
[11-20]	63,981	37,562	6,752,150	3,751,124
[21-30]	17,710	10,567	5,374,523	3,185,935
[31-40]	10,863	4,461	6,666,546	2,646,306
[41-50]	6,318	2,504	6,281,387	2,502,691
[51-60]	2,663	1,300	3,978,483	1,972,822
[61-70]	1,378	832	2,891,198	1,750,014
[71-80]	1,058	682	2,960,532	1,899,880
[>80]	4,978	3,999	137,488,518	113,935,797

Offer categorization. E-shops use a wide range of different categorization schemata to present their offers. In order to consistently categorize all offers into the same set of product categories, we apply transfer learning similar to [14]. Using a publicly available amazon.com dataset of product reviews and metadata[7], we build lexica containing terms and their TF-IDF scores for 26 product categories. For every offer in the training set we calculate a score for each category by considering the overlapping terms after TF-IDF weighting between the category lexicon and the schema.org properties *name, title, description*, and *brand* of the offer entity. The offer is assigned the category with the maximum score. In case the term overlap is minimal, the offer is assigned the category label "not found". Finally, we use majority voting among the offers of a cluster to assign cluster specific category labels. Thus, all offers belonging to one cluster are labeled with the same product category.

3 TRAINING DATASET PROFILING

This section analyses the structure of the WDC Training Dataset for Large-Scale Product Matching as well as the structure of its English-language subset. The English-language subset is created by selecting all offers from pages having the suffixes: com, net, co.uk, and org. The Full training set contains 26 million offer entities, deriving from 79 thousand websites, grouped into 16 million clusters. The English training set contains 16 million offer entities, deriving from 43 thousand websites, grouped into 10 million clusters. Table 3 shows the distribution of offer entities per cluster in the Full and English training sets as well as the amounts of positive pairs. We observe that small clusters (size one and two) account for 92% of the clusters in the Full training set. The reasons for the large fraction of small clusters are twofold: First, the long tail distribution of products on the Web. Second, the limited depth of the CommonCrawl as only a fraction of the pages of each website is crawled. However, the English training set contains over 600 thousand clusters having a size of three or larger.

[7] http://jmcauley.ucsd.edu/data/amazon/

Product Categories in the English Training Set.

Table 4 shows the distribution of clusters per product category as well as the clusters size distribution in the English training set. Considering clusters of size three and larger, the positive pairs for the categories *Shoes, Camera and Photo, Cell Phone and Accessories, Computers and Accessories,* and *Jewelry* are more than 250 thousand while for the categories *Office Product* and *Clothing* there are more than 1 million positive pairs.

Table 4: Distribution of product categories in the English training set

Category	% Clusters	# Clusters of Size			
		[3-4]	[5-10]	[11-20]	[>20]
Office	10.90	40,314	16,920	5,953	3,043
Jewelry	7.79	33,156	13,329	3,352	2,037
Clothing	6.80	49,085	30,384	3,285	1,866
Automotive	5.90	16,650	8,139	2,865	2,140
Beauty	5.78	27,636	10,568	2,115	1,070
Phones & Acc.	4.76	15,870	5,162	1,085	878
Home & Kitchen	4.68	24,429	7,414	1,247	538
Luggage	4.48	14,401	6,399	1,198	957
Tools	4.35	12,033	4,407	1,248	1,042
CDs & Vinyl	4.19	17,666	6,013	1,417	663
Shoes	4.11	16,603	7,590	1,335	721
Camera & Photo	3.47	14,583	5,408	1,423	935
Grocery	3.26	17,109	5,889	2,154	716
Computers & Acc.	3.20	11,614	5,411	2,308	2,862
not found	3.07	12,964	4,088	942	267
Digital Music	3.03	7,954	3,046	640	535
Other Electronics	2.83	11,649	4,412	977	427
Books	2.81	9,889	2,946	330	183
Video Games	2.62	8,256	3,419	938	779
Garden	2.43	4,898	1,764	475	366
Musical Instr.	2.31	5,182	1,684	550	486
Pet Supplies	2.15	7,605	2,974	620	440
Baby	1.71	5,509	1,894	458	254
Toys	1.19	3,120	1,016	258	189
Sports	0.75	3,460	1,234	314	372
Movies & TV	0.71	2,030	681	195	157
Health	0.70	6,857	1,165	189	113

Property distribution. Table 5 shows the distribution of the schema.org properties that are used to describe the offers in the Full and English training sets. In addition, we report the distribution of identifier related schema.org properties in both sets. Our specification table detection method finds at least one specification table in 24% of the HTML pages contained in the Full set and 17% of the pages of the English set. Using the key/value pair extraction heuristic described in Section 2, we are able to extract ten or more key/value pairs from 73% of the specification tables. Finally, 80% of the offers in both the Full and English training sets contain the *dcterms:title* property of the DCMI Type Vocabulary[8] which captures the content of the HTML <title> element.

Table 5: schema.org properties in the Training Dataset and Gold Standard

Property	Offers Full		Offers English		Gold	
	# (in k)	%	# (in k)	%	#	%
name	25,281	95.3	15,653	95.1	2,300	99.6
description	17,215	64.9	11,352	69.0	1,884	81.6
brand	9,313	35.1	5,645	34.3	767	33.2
image	5,785	21.8	4,348	26.4	407	17.6
price	3,335	12.5	1,977	12.0	301	13.0
priceCurr.	2,971	11.2	1,873	11.3	293	12.6
availability	1,180	4.4	716	4.3	170	7.3
manufact.	2,024	7.6	1,254	7.6	325	14.0
sku	11,475	43.2	7,239	44.0	747	32.3
mpn	4,611	17.3	3,167	19.2	1,504	65.1
productID	9,386	35.4	6,351	38.6	348	15.0
gtin8	452	1.7	167	1.0	130	5.6
gtin13	3,529	13.3	1,449	8.8	263	11.3
gtin12	300	1.1	261	1.5	16	0.6
gtin14	420	1.5	71	0.4	9	0.3
identifier	179	0.6	65	0.3%	5	0.2

4 WDC GOLD STANDARD FOR LARGE-SCALE PRODUCT MATCHING

Due to the general noisiness of web data as well as anomalies not resolved by our cleansing procedure, there is no guarantee that all offers in a cluster will always refer to the same product or that products in different clusters are always different. In order to allow matching methods to be evaluated on completely clean data, we create the WDC Gold Standard by manually verifying for a set of 2200 pairs of offers whether they refer to the same product or not.

The difficulty of a matching task as well as the suitable matching method for a task both depend on the structuredness of the data to be matched. Thus, we select two product categories containing less structured offers (watches and sneaker shoes) and two categories containing more structured offers (computers & accessories and camera & photo) and create one gold standard for each using offer pairs from our corpus. First, we identify the clusters belonging to the selected product categories. We select 75 related clusters preferring clusters having a large diversity among the offers' textual content and a minimum size of 5 offers. Large diversity in this context refers to offers for the same product that vary on the Jaccard string similarity of their titles and descriptions, thus leading to a selection of clusters that contain textually similar as well as less similar offers.

In order to select challenging pairs of offers for the manual verification, we apply the following procedure: From every selected cluster we pick one offer and exploit its textual content given by the dcterms:title, schema:name, schema:description, and specification table values. Similarly to [9] we use the Jaccard similarity metric and the offers' textual content to calculate the similarity scores between the picked offer and the offers of the same cluster (intra-cluster similarity scores) as well as the offers of different clusters from the full corpus(inter-cluster similarity scores). We select the intra-cluster offer pairs with the highest and lowest similarity scores and add them in the gold standard. In addition, we add one to

[8] http://dublincore.org/documents/2008/01/14/dcmi-type-vocabulary/

three inter-cluster offer pairs with the highest similarity score and three randomly chosen inter-cluster pairs in the gold standard. We manually verify that the selected pairs are really matches or non-matches by reading the textual content of the offers. If we discover that a pair is incorrectly labeled, we correct the label. Finally, we remove the manually annotated pairs from the training set.

The resulting gold standard datasets consist of 150 positive and 400 negative pairs for each category. The offers contained in the gold standard datasets originate from the following numbers of clusters for each category: 338 for Computers & Accessories, 231 for Camera & Photo, 269 for Watches and 186 for Sneakers. The two right-most columns in Table 5 describe the density of the properties of the offers contained in the gold standard.

5 BASELINE IDENTITY RESOLUTION EXPERIMENTS

This section presents a set of matching experiments conducted using the English training set and the WDC gold standard. The experiments are intended on the one hand to verify the utility of the WDC training set as well as the cleansing procedure that was used to create the training set. On the other hand, we use the training set and gold standard to publicly replicate the results of Mudgal et al. [12]. First, we perform an unsupervised bag-of-words experiment using TF-IDF and cosine similarity. Afterwards, we train various supervised models such as logistic regression, naive Bayes, LinearSVC, decision trees, and random forests using (i) binary word co-occurrence vectors and (ii) string similarity scores, automatically generated by the Magellan framework [8], as features. As neural network based matchers, we combine all network types implemented in the *deepmatcher* framework (e.g. RNNs, Attention, and Hybrid) with pre-trained and self-trained *fastText* embeddings.

We experiment with different subsets of the offer features title, description, brand, and specification table content. For the title feature we concatenate the textual values schema:name and dcterms:title. All identifier related properties (lower part of the Table 5) are removed from the offers. Due to resource limitations, we do not use the complete English training set for the supervised experiments but subsets of potentially interesting training examples (e.g. positive pairs from many different clusters and negative pairs from different clusters where both offers have a rather similar description). Table 6 gives an overview of the size (number of positive and negative pairs), diversity (number of origin clusters) and feature density. We abbreviate the features schema:name and dcterms:title with T, schema:description with D, schema:brand with B and specification tables with S.

The results of the experiments are summarized in Table 7. For each category, we report the best performing method/feature combinations. As expected the supervised methods outperform the unsupervised BOW approach significantly. More interestingly, the deep learning approaches using *fastText* embeddings are 5-11% better in F1 compared to the supervised methods using symbolic feature representations. This confirms the result of Mudgal et al. that deep learning based matching methods excel on tasks involving rather textual entity descriptions. More information about the exact configuration of all methods as well as the results of method/feature combinations of weaker performance are found on the project's website.

Table 6: Training set profiling

Category	# Pos.	# Neg.	Clusters	% Feat. Density		
	Pairs	Pairs		T	D	S
Computers	20,444	21,676	338	100	82	55
Cameras	7,539	9,093	231	100	61	4
Watches	5,449	8,819	269	100	48	4
Shoes	3,476	5,924	186	99	36	1

Table 7: Results of all experiments

Category	Classifier	Features	P	R	F1
Unsupervised Matching					
Computers	Cosine, TF-IDF	T+D+B	0.52	0.70	0.60
Cameras	Cosine, TF-IDF	T+D+B	0.52	0.83	0.64
Watches	Cosine, TF-IDF	T	0.45	0.89	0.60
Shoes	Cosine, TF-IDF	T	0.61	0.76	0.67
Supervised Matching - Symbolic Features					
Computers	LinearSVM	T+D	0.75	0.94	0.84
Cameras	LinearSVM	T+D+B+S	0.70	0.87	0.78
Watches	LinearSVM	T+D+B+S	0.74	0.91	0.81
Shoes	LinearSVM	T+D+B+S	0.72	0.95	0.82
Computers	RandomForest	T	0.72	0.92	0.81
Cameras	RandomForest	T+D+B+S	0.75	0.87	0.81
Watches	RandomForest	T+D+B	0.66	0.91	0.77
Shoes	RandomForest	T+D+B+S	0.67	0.95	0.79
Supervised Matching - Symbolic Features - Magellan					
Computers	RandomForest	T+D	0.59	0.79	0.67
Cameras	RandomForest	T+D+B+S	0.53	0.85	0.65
Watches	RandomForest	T+D+B+S	0.71	0.85	0.78
Shoes	RandomForest	T+D+B+S	0.71	0.95	0.81
Supervised Matching - Distributed Features - DeepMatcher					
Computers	RNN	T+D+B+S	0.84	0.96	**0.89**
Cameras	RNN	T+D+B+S	0.88	0.93	**0.90**
Watches	RNN	T+D+B+S	0.88	0.97	**0.92**
Shoes	RNN	T+D+B+S	0.88	0.97	**0.92**

6 RELATED WORK

This section compares the *WDC Training Dataset for Large-Scale Product Matching* with existing resources for the evaluation of entity resolution methods. There exist various evaluation datasets for the task of product matching. The two classic datasets in this area *Abt-Buy* and *Amazon-Google* were introduced by Köpcke and Rahm [10]. Gokhale et al. introduce another public product dataset *Walmart-Amazon* [5]. Mudgal et al. [12] use several large product datasets for evaluating their deep learning methods. Unfortunately, these datasets are not public. Various benchmark datasets have been introduced for the Instance Matching Track of the Ontology Alignment Evaluation Initiative (OAEI)[9] over the years. Daskalaki et al. give an overview of these datasets [4]. The 2017 OAEI Instance Matching Track used the evaluation datasets SYNTHETIC and

[9] http://oaei.ontologymatching.org/

Table 8: Comparison of evaluation datasets for entity resolution

Dataset	Public	# Data Sources	# Positive Pairs
Walmart-Amazon [5]	yes	2	1,154
Amazon-Google [10]	yes	2	1,300
Abt-Buy [10]	yes	2	1,097
DBLP-ACM [10]	yes	2	2,224
DBLP-Scholar [10]	yes	2	5,347
DM-Clothing [12]	no	1	105,608
DM-Electronics [12]	no	1	98,401
DM-Home [12]	no	1	111,714
DM-Tools [12]	no	1	96,836
DM-Company [12]	yes	?	28,200
OAEI - SYNTHETIC [1]	yes	1	1,800
Citeceer - DBLP [2]	yes	2	558,787
Falcon - Songs [3]	yes	1	1,292,023
WDC - Product GS [15]	yes	32	1,500
WDC - LSPM	yes	79,126	40,582,671
WDC - LSPM English	yes	43,293	20,773,304

DOREMUS [1]. A large citation dataset *Citeseer - DBLP* offering 550 thousand matches is provided in the Magellan Data Repository [2]. A large song dataset containing 1.2 million matching pairs has been used to evaluate Falcon [3]. As part of our previous work [15], we have published a gold standard for product data extraction and matching covering 32 different e-shops.

Table 8 compares the *WDC Training Dataset for Large-Scale Product Matching (WDC - LSPM)* to other evaluation datasets along the dimensions number of positive pairs (e.g. offers referring to the same product) as well as number of sources from which the data originates. The table shows that concerning the number of positive pairs WDC - LSPM is several orders of magnitude larger than the existing evaluation datasets in the area of product matching (public as well as proprietary datasets). Compared to the Falcon-Songs dataset, WDC - LSPM English is 17 times larger. Concerning the number of sources, WDC - LSPM English covers 43,293 sources while the existing datasets cover at most 32 sources.

7 CONCLUSION

The *WDC Training Dataset* nicely demonstrates the utility of the Semantic Web. Without the website owners putting semantic annotations into their HTML pages it would have been much harder, if not impossible, to extract product offers from 79 thousand e-shops. While the training set likely still contains some noise that the cleansing procedure did not remove, being able to achieve F1 scores around 0.90 in the experiments clearly demonstrates the utility of the training set. We hope that researchers working on entity resolution and e-commerce will consider the *WDC Training Dataset and Gold Standard* useful and we hope that both artefacts contribute to advance the understanding of the potentials of latent semantic representations and deep neural networks for the task of product matching.

REFERENCES

[1] Manel Achichi, Michelle Cheatham, et al. 2017. Results of the Ontology Alignment Evaluation Initiative 2017. In *Proceedings of OM 2017-12th ISWC workshop on ontology matching*. 61–113.

[2] Sanjib Das, AnHai Doan, et al. 2016. The Magellan Data Repository. https://sites.google.com/site/anhaidgroup/useful-stuff/data.

[3] Sanjib Das, Paul Suganthan G.C., et al. 2017. Falcon: Scaling Up Hands-Off Crowdsourced Entity Matching to Build Cloud Services. In *Proceedings of the 2017 ACM International Conference on Management of Data (SIGMOD '17)*. 1431–1446.

[4] Evangelia Daskalaki, Giorgos Flouris, et al. 2016. Instance Matching Benchmarks in the Era of Linked Data. *Journal of Web Semantics* 39 (2016), 1 – 14.

[5] Chaitanya Gokhale, Sanjib Das, et al. 2014. Corleone: Hands-off Crowdsourcing for Entity Matching. In *Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data (SIGMOD '14)*. 601–612.

[6] Anitha Kannan, Inmar E. Givoni, et al. 2011. Matching Unstructured Product Offers to Structured Product Specifications. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '11)*. 404–412.

[7] Elias Kärle, Anna Fensel, et al. 2016. Why Are There More Hotels in Tyrol than in Austria? Analyzing Schema.org Usage in the Hotel Domain. In *Information and Communication Technologies in Tourism 2016*. Cham, 99–112.

[8] Pradap Konda, Jeff Naughton, and et al. 2016. Magellan: toward building entity matching management systems. *Proceedings of the VLDB Endowment* 9, 12 (2016), 1197–1208.

[9] Hanna Köpcke and Erhard Rahm. 2008. Training selection for tuning entity matching.. In *Proceedings of the 6th International Workshop on Quality in Databases and Management of Uncertain Data (QDB/MUD '08)*. 3–12.

[10] Hanna Köpcke, Andreas Thor, and Erhard Rahm. 2010. Evaluation of Entity Resolution Approaches on Real-world Match Problems. *Proceedings of the VLDB Endowment* 3, 1-2 (2010), 484–493.

[11] Robert Meusel and Heiko Paulheim. 2015. Heuristics for Fixing Common Errors in Deployed Schema.Org Microdata. In *Proceedings of the 12th European Semantic Web Conference on The Semantic Web. Latest Advances and New Domains - Volume 9088*. 152–168.

[12] Sidharth Mudgal, Han Li, et al. 2018. Deep Learning for Entity Matching: A Design Space Exploration. In *Proceedings of the 2018 International Conference on Management of Data (SIGMOD '18)*. 19–34.

[13] Petar Petrovski and Christian Bizer. 2017. Extracting Attribute-value Pairs from Product Specifications on the Web. In *Proceedings of the International Conference on Web Intelligence (WI '17)*. 558–565.

[14] Petar Petrovski, Volha Bryl, and Christian Bizer. 2014. Integrating Product Data from Websites Offering Microdata Markup. In *Companion Proceedings of the 23rd International Conference on World Wide Web (WWW '14 Companion)*. 1299–1304.

[15] Petar Petrovski, Anna Primpeli, et al. 2017. The WDC Gold Standards for Product Feature Extraction and Product Matching. In *E-Commerce and Web Technologies*. 73–86.

[16] Disheng Qiu, Luciano Barbosa, et al. 2015. Dexter: Large-scale Discovery and Extraction of Product Specifications on the Web. *Proceedings of VLDB Endowment* 8, 13 (2015), 2194–2205.

[17] Petar Ristoski, Petar Petrovski, et al. 2018. A machine learning approach for product matching and categorization. *Semantic Web* 9, 5 (2018), 707–728.

[18] Kashif Shah, Selcuk Kopru, et al. 2018. Neural Network based Extreme Classification and Similarity Models for Product Matching. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 3 (Industry Papers)*. 8–15.