MWPD2020: Semantic Web Challenge on Mining the Web of HTML-embedded Product Data

Ziqi Zhang[1][0000−0002−8587−8618], Christian Bizer[2][0000−0003−2367−0237], Ralph Peeters[2][0000−0003−3174−2616], and Anna Primpeli[2][0000−0002−1783−2482]

[1] University of Sheffield, Broomhall, Sheffield S10 2TG, UK
ziqi.zhang@sheffield.ac.uk
[2] Universität Mannheim, Schloss, 68131 Mannheim, Germany
{chris,ralph,anna}informatik.uni-mannheim.de

Abstract. This paper gives an overview of the Semantic Web Challenge on Mining the Web of HTML-embedded Product Data (MWPD2020) which has been conducted as part of the International Semantic Web Conference (ISWC2020). The challenge consists of two tasks: product matching and product classification. In the first task, participants need to identify offers for the same product originating from different websites. The goal of the second task is to categorize offers from different websites into the GS1 GPC product hierarchy. Six teams from the USA, China, Japan, and Germany participated in the challenge. The winning system in Task 1, PMap, achieved an F1 score of 86.05 using an ensemble of transformer-based language models. Task 2 was won by team Rhinobird achieving a weighted average F1 score of 88.62 using a BERT-based ensemble which considers the dependencies among different category levels.

Keywords: entity matching · hierarchical classification · e-commerce · schema.org · microdata · benchmarking

1 Introduction

Recent years have seen significant growth of semantic annotations on the Web, using markup languages such as Microdata together with the schema.org vocabulary. A particular domain that is witnessing the boom of semantic annotations is e-commerce, where online shops are increasingly embedding schema.org annotations into HTML-pages describing products in order to enable search engines to easily identify product offers and potentially drive traffic to the respective websites. Statistics from the Web Data Commons (WDC) project[3] show that, as of November 2018, 37% of web pages, or 30% of websites contain semantic annotations, amounting to over 30 billion facts. Among these, nearly 20% are

Copyright © 2020 for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

[3] http://webdatacommons.org/structureddata/

CEUR-WS.org/Vol-2720/paper1.pdf

related to products. Such structured product data on the Web have created opportunities for new services, such as product search and integration platforms, recommender systems [1], as well as emerging research fields such as product knowledge graphs [6].

Many websites have started to semantically annotate product identifiers within their pages. This enables the identification of offers for the same product on different websites. The resulting clusters of product descriptions can be used as weak supervision for training product matchers, which in turn can be applied to identify products on websites that do not provide product identifiers [3]. However, there are also challenges associated with the annotations. For example, less than 10% of the offers are annotated with a product category [16], while the used categorization systems are website-specific and highly inconsistent across different websites [22].

The potentials as well as the challenges resulting from the wide-spread availability of semantically annotated product data on the Web motivated the Semantic Web Challenge on Mining the Web of HTML-embedded Product Data (MWPD2020), as well as the specific tasks of the challenge: product matching and product classification. In the first task, participants need to identify offers for the same product originating from different websites. The goal of the second task is to categorize offers from different websites into the GS1 GPC product hierarchy. For both tasks, we have assembled training, validation, and test sets consisting of semantically annotated product data from a wide variety of different websites.

The event attracted a total of six participating teams including research institutions as well as commercial entities from the USA, China, Japan, and Germany. The winning team for the product matching task represents the National Institute of Informatics, Japan; while the winning team for the product classification task represents the Tongji University of China and Tencent, China.

The remainder of this paper is structured as follows. Section 2 and 3 explain the two tasks, including their objectives, datasets, and evaluation metrics; Section 4 presents the results of the challenge and gives an overview of the participating systems; and Section 5 concludes the paper with some lessons learned from the challenge and a comparison of MWPD2020 to related benchmark events. More information about MWPD2020 is found on the challenge's website which also provides all datasets for public download[4].

2 Task 1 - Product Matching

E-commerce websites frequently annotate product identifiers, product titles, product descriptions, brands, and product prizes within their pages using schema.org terms. In addition, offers are often accompanied by specification tables, i.e. HTML tables that contain product details in the form of key/value pairs. Given the syntactic, structural and semantic heterogeneity among the offers, it is challenging to identify which offers refer to the same product, a problem known as

[4] https://ir-ischool-uos.github.io/mwpd/

product matching. In this task, product matching is handled as a binary classification problem: given two product offers, the participating systems need to decide whether the offers describe the same product (matching) or not (non-matching).

2.1 Datasets

The participants of Task 1 were given a large corpus of product offers which are grouped into clusters referring to the same product. This corpus could be used by the participants to assemble training sets of different width and depth. In order to ease starting to work on the task, we also provide a readily assembled example training and validation set. The test set that was used to evaluate the participating systems was kept secret during the submission period of the challenge and was released afterwards.

Product Data Corpus. The WDC Product Data Corpus [20] was released in 2018[5] by the Web Data Commons project and is the largest publicly available product data corpus. It consists of 26 million product offers originating from 70 thousand different e-shops. Exploiting the weak supervision found on the web in the form of product identifiers, such as GTINs or MPNs, product offers are grouped into 16 million clusters. The clusters can be used to derive training sets containing matching and non-matching pairs of offers. The grouping of offers into clusters is subject to some degree of noise which is approximately 7%. The following attributes are used for describing the product offers in the corpus and can be used for training: *title*, *description*, *brand*, *price*, *specTableContent* which contains the content of the specification tables found in the website of the product offer, *keyValuePairs* which are the heuristically extracted key/value pairs from the specification tables and category which is one of the 25 categories the offer was assigned to. Additionally, two identifier attributes are assigned to every product offer: the *id* which is the unique identifier of the offer and the *cluster_id* which is the identifier of the cluster to which the offer belongs.

Example Training Set. Generating interesting matching and non-matching pairs of offers which can be used for training powerful matching models, is a non-trivial and resource intense task. Therefore, we offer an example of a training set derived from the product corpus, with the goal to additionally support the participants of this task. Being a direct subset of the product corpus, the example training set is subject to some inherent noise. The example training set contains 68K pairs of matching and non-matching offers from 772 distinct products (clusters of offers). We offer the example training set in JSON format. Every JSON object in the training set describes a pair of offers (left offer - right offer) using the offer attributes together with their corresponding matching label. Figure 1 shows an example of a non-matching product offer pair in the example training set.

[5] http://webdatacommons.org/largescaleproductcorpus/v2/

```
{
   "category_left":"Computers_and_Accessories",
   "category_right":"Computers_and_Accessories",
   "cluster_id_left":8664,
   "cluster_id_right":577633,
   "id_left":12491611,
   "id_right":16963082,
   "label":0,
   "pair_id":"12491611#16963082",
   "brand_left":"WD",
   "brand_right":"Intel",
   "description_left":"6TB Capacity, SATA 6GB/s Interface, 3 Year Warranty",
   "description_right":"Core i5-7600K 3.8 GHz Quad-Core LGA 1151 Intel",
   "keyValuePairs_left":{"Product Measures":"3.7cm x 3.7cm x 1cm"},
   "keyValuePairs_right":{"Genre ":"CPU"},
   "price_left":"24.50",
   "price_right":"241.99",
   "specTableContent_left":"Product Measures 3.7cm x 3.7cm x 1cm",
   "specTableContent_right":"Genre CPU",
   "title_left":"WD Red 6TB 5400rpm SATA 6Gb/s 64MB",
   "title_right":"Intel Core i5-7600K, 4x 3.80GHz"
}
```


Fig. 1. Example of a product offer pair in the training set.

Validation Set. We provide a validation set consisting of 1,100 offer pairs from the Computers and Accessories category as the ground truth for this task. The validation set has the same structure as the example training set. The ratio of matching to non-matching pairs is 3:8. The offers of the validation set are derived from 745 distinct products (clusters). Table 1 presents for the training and validation sets, the average attribute density, the average length in characters of the attribute values, as well as the standard deviation of the value length.

Table 1. Profiling statistics about the offers in the training and validation sets.

Attribute	Density	Avg Length	Std Length
title	1.000	90.14	30.89
description	0.743	366.6	1167.76
brand	0.533	12.43	4.8
price	0.206	15.18	3.85
specTableContent	0.327	557.21	676.48
keyValuePairs	0.298	497.96	429.02

With the aim to construct the validation set using a good mixture of hard and easy matching and non-matching pairs of offers distributed over different products, we applied the following heuristic: First 150 clusters of the category Computers and Accessories are randomly selected. Considering the clustering scheme, we construct all possible matching and non-matching pairs. For each offer pair we randomly pick the Jaccard similarity over the titles, the descriptions

or the average of both, as a similarity metric and calculate its similarity score. To select matching pairs, we pick within each cluster the pair with the lowest similarity score and one randomly chosen pair and add them in the validation set. To select negative pairs, we take for each offer two to three pairs with high similarity and three pairs at random. All matching and non-matching pairs of the validation set are manually verified. Therefore, unlike the provided example training set, the validation set does not contain any noisy offer pairs.

Test Set. The hidden test set which is used for evaluating and ranking the matching systems of the participants of this task, consists of 1,500 offer pairs from the category Computers and Accessories. The offer pairs in the test set are carefully selected in order to cover different types of matching challenges. 1,100 pairs are randomly selected corner cases, meaning that they are similar non-matching pairs and dissimilar matching pairs. For the remaining 400 pairs, we define a categorization scheme consisting of four specific types of matching challenges. The distribution of offer pairs per type of challenge remains unknown to the participants in the first round and is revealed to them at the beginning of the second round to allow them to tune their systems to the specific challenges. The distribution of offer pairs per type of matching challenge is shown in Table 2.

Table 2. Distribution of offer pairs per type of matching challenge.

Matching challenge	#Match	#Non-Match
(SN-DM) Similar non-matches, dissimilar matches	275	825
(NP-HS) New products - high similarity to known products	25	75
(NP-LS) New products - low similarity to known products	25	75
(KP-TY) Known products - introduced typos	100	0
(KP-DR) Known products - dropped tokens	100	0

With the term *new product* we refer to products which are contained in the WDC Product Data Corpus but not in the provided example training set. *Known products* means products from clusters that have training data in the provided training set. The similarity for choosing similar non-matching and dissimilar matching pairs is measured using the Jaccard similarity metric on the titles of the offers.

2.2 Evaluation Metrics and Baseline

For the evaluation of the product matching task we use standard precision, recall and F1 calculated on the positive class. As a baseline, we use the Deepmatcher [17] framework, more specifically the RNN module using default parameters apart from the positive/negative ratio, which we set to the actual distribution found in the training set. This model is trained on the training set provided for the challenge for 15 epochs, using the attributes *title*, *description*,

brand and *specTableContent*. We preprocess the attributes by removing some symbols and schema.org related terms and finally lowercasing them. Since the model relies on pre-trained word- or character-based embeddings, we use the fastText embeddings pre-trained on the English Wikipedia[6].

3 Task 2 - Product Classification

Same products are often sold on different websites, which generally organise their products into certain categorisation systems. However, such product categorisations differ significantly for different websites, even if they sell similar product ranges. This makes it difficult for product information integration services to collect and organise product offers on the Web. The product classification task deals with assigning pre-defined product category labels from a universal catalogue to product instances (e.g., iPhone X is a 'SmartPhone', and also 'Electronics').

3.1 Datasets

Classification labels. In this task, the GS1 Global Product Classification standard (GPC) [7] is used to classify product instances. The GPC standard classifies products into a hierarchy of multiple levels based on their essential properties as well as their relationships to other products. It offers a universal standard for organising product catalogues of any businesses. In this task, the top three levels of GPC are used to classify each product. Level 1 contains 40 classes such as 'Automotive' and 'Clothing'. Level 2 further divides level 1 classes into more than 100 classes such as 'Automotive Accessories and Maintenance', and 'Swimwear'. Then level 3 further divides level 3 classes into over 700 classes such as 'Automotive Antifreeze' and 'Beachwear/Cover Ups'

Gold standard. The creation of the gold standard is based on the earlier product classification dataset released by the Web Data Commons project [16]. The original dataset contained 8,361 product instances randomly sampled from 702 vendors' websites. These were manually classified into the above mentioned three levels of classifications by human annotators.

In MWPD2020, we further extended the original GS by adding over 7,000 product instances. However, due to the complexity of the GPC hierarchy, annotating a random sample by checking against every class in the three classification levels is a very time-consuming process. Therefore, we followed a 'controlled process' detailed below.

1. Create a Solr[8] index of product instances by parsing the Product Data Corpus. The index contains five fields corresponding to attributes of each product: an *id* field to uniquely identify a product index; a *name* field recording

[6] https://fasttext.cc/docs/en/pretrained-vectors.html
[7] https://www.gs1.org/standards/gpc
[8] https://lucene.apache.org/solr/

the name of the product; a *description* field recording the long description of the product; a *category text* field recording the product category information as provided by the source web page; and a *provenance* field recording the source URL from which the structured data are extracted. All these attributes are extracted from the RDF quads where available in the dataset.
2. Given an existing product instance (i.e., the reference product) in the original GS, search for its name in the *description* field of the above index;
3. Select up to 50 results (i.e., the target products) with a different *name* from the reference product;
4. Rank the results by the Levenshtein distance between the reference product's name and the names of the target products;
5. Select the top 10 ranked results;
6. A human annotator manually evaluates the ranked results, and selects `n` products that s/he deems to belong to the same level-3 class of the reference product;
7. The selected `n` products are assigned the same level-3 class, as well as the corresponding level-2 and level-1 classes by traversing the GPC hierarchy.

In step 6 above, the human annotators are presented the following information when assessing each target product: the reference product's name, description, level-1, 2, 3 classes, provenance, website-specific category information or breadcrumb if available; and the target product's name, description and provenance. They are instructed to exercise on their own discretion to decide an optimal `n`, by balancing the diversity in the already selected selected target products in terms of their name, vendor as identified by their provenance, and level-3 classes.

The main reasons of allowing this flexibility are two-fold. First, in the original GS, there existed certain 'difficult' and often minority classes (e.g., `93070100_Seeds/Spores`) for which steps 2 and 3 hardly returned many positive matches. Second, there also existed certain 'dominant' classes that represented a very large fraction of the original GS (e.g., `67000000_Clothing` was over 40%) and were also 'easier' to find matches by steps 2 and 3. This implies that our controlled process runs a risk of further accentuating the already-unbalanced nature of the original GS. Thus by exercising their discretion based on the above principle, our goal was to control the balance in the distribution of classes in the end dataset. In practice, `n` ranged between 0 (no suitable target) and 6 (typically for 'difficult' classes).

The annotation was conducted by two computer science researchers and Inter-Annotator-Agreement was studied on 100 product instances where they both annotated. A Cohen's Kappa of 97% was obtained. The end dataset contains 16,119 instances and is stored in JSON. In addition to the five product attributes described before, each instance is assigned three label, corresponding to for the three levels of classification. Further, the description is truncated to a maximum of 5,000 characters. A screenshot of an example instance is shown in Figure 2. The dataset is split into the training (10,012), validation (3,000), and test (3,107) sets with statistics shown in Figures 3 and 4. Same as Task

1, only the training and validation sets are revealed to the participants before the submission of their final system output that was created on the test set. Although the dataset are very unbalanced, with several large classes dominating the dataset at all three levels, it is worth noting that they follow a consistent distribution to the original GS.


```
{"ID": "3307",
        "Name": "East Carolina Pirates Ladies Personalized Basketball Long Sleeve Classic Fit T-Shirt
        "Description": "Feel like a bona fide member of East Carolina Pirates athletics by sporting y
        "CategoryText": "East Carolina Pirates > East Carolina Pirates Ladies > East Carolina Pirates
        "URL": "http://eastcarolina.teamfanshop.com/COLLEGE_East_Carolina_Pirates_Ladies_Long_Sleeve",
        "lvl1": "67000000_Clothing",
        "lvl2": "67010000_Clothing",
        "lvl3": "67010800_Upper Body Wear/Tops"}
```


Fig. 2. Example of an instance in the Task 2 Product Classification dataset.

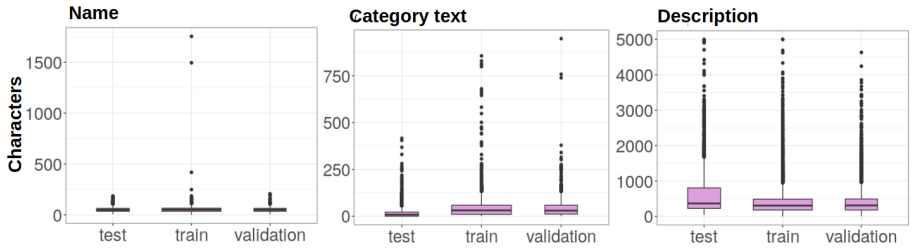

Fig. 3. Distribution of the character lengths of product names, category texts, and descriptions in the training, validation and test sets.

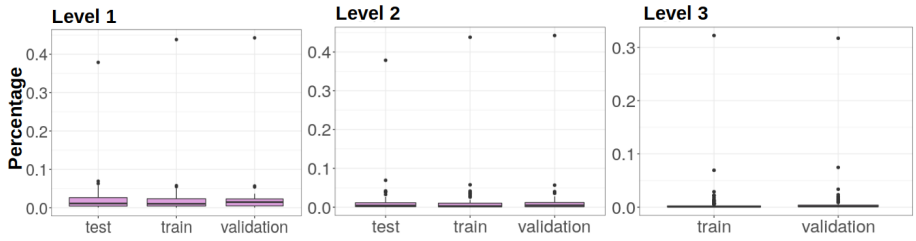

Fig. 4. Distribution of the percentages of level-1, 2, and 3 classes in the training, validation and test sets.

Additional resources. During the process of creating the gold standard, additional resources[9] were created with an aim support participants system development. These include:

- A 'product data textual corpus' that contains descriptions of all product instances from the Solr index above. A light-weight cleaning process was applied to only keep descriptions of at least 5 tokens (separated by white-space characters) and 20 characters. This corpus has over 1.9 billion tokens.
- Word embedding models (both continuous Bag-of-Words (CBow) and Skip-gram) trained using the above textual corpus, by applying the Gensim[10] (version 3.4.0) implementation of Word2Vec.

3.2 Evaluation Metrics and Baseline

For each classification level, the standard Precision, Recall and F1 are used and a Weighted-Average macro-F1 (WAF1) are calculated over all classes. Then the average of the WAF1 of the three levels are be calculated and used to rank the participating systems. For baseline, we use a configuration based on that used in the Rakuten Data Challenge[11] [14]. Specifically:

- it uses the same FastText algorithm and parameters as in the Rakuten Data Challenge;
- it uses only product names as input text;
- all text are lowercased and lemmatised using NLTK version 3.4.5.

4 Results

The competition was organised in two rounds. In order to improve their systems, the teams were shown the leaderboard after Round 1 and were informed about the F1 scores that their systems achieved on the specific types of matching challenges (see Table 2). A total of six teams representing different industry organisations and research and academic institutions participated in Round 1. Six teams participated in Task 1, while five participated in Task 2. Two teams continued in Round 2 for Task 1, while only one team chose to continue in Round 2 for Task 2. Five teams submitted a paper describing their system. A brief overview of the results for both tasks is given below. Section 4.3 provides additional details about each team and their system afterwards.

4.1 Results Task 1 - Product Matching

Table 3 shows the results of the systems that participated in Task 1. Team PMap managed to achieve the highest F1-score among all teams with teams Rhinobird

[9] Download from: https://bit.ly/36d0NYd
[10] https://radimrehurek.com/gensim/
[11] Download from: https://github.com/ir-ischool-uos/mwpd/tree/master/prodcls

and ISCAS-ICIP very close behind. An overview of each team's methods along selected dimensions which resulted in the final submission can be seen in Table 4. All of the teams, who submitted system papers, employed fine-tuning of transformer-based pre-trained language models for the task of product matching in one form or another, often combined with some form of ensembling. Team ISCAS-ICIP was the only team employing an ensembling approach across different deep matching models. The other teams, who employed ensembling, limited themselves to fine-tuned transformer-based models like e.g. BERT [5] and RoBERTa [15].

Team	P	R	F1
PMap	82.04	90.48	**86.05**
Rhinobird (Round 2)	80.63	92.00	85.94
Rhinobird (Round 1)	82.86	88.38	85.53
ISCAS-ICIP (Round 2)	85.77	84.95	85.36
ASVinSpace	86.20	82.10	84.10
ISCAS-ICIP	83.89	81.33	82.59
Megagon	82.69	65.52	73.11
Team ISI	78.44	57.52	66.37
Baseline (Deepmatcher)	70.89	74.67	72.73

Table 3. Results of Task 1 - Product Matching.

Most teams applied some form of standard pre-processing to the data before training. Team ISCAS-ICIP also employs a feature extraction approach based on fixed vocabularies and regular expressions to extract multiple features from the textual descriptions. For this they use the provided WDC Large-Scale Corpus for Product Matching. They further use it to expand the provided training set with more product pairs from the relevant product category. Also, teams Rhinobird and ASVinSpace tried augmenting the training data with pre-built training sets of other categories from the product corpus. Most teams tried different combinations of features and training sets during experimentation. Team Rhinobird also tried optimizing for focal loss in addition to cross-entropy as well as employing a self-ensembling technique over multiple training epochs of the same model. Rhinobird and ISCAS-ICIP further implemented a post-processing step, where they used heuristic rules to correct some of the predicted labels, e.g. always predicting *non-match* if the brands of both offers do not match.

All teams were provided with information about their performance on the specific matching challenges in the test set (see section 2) after Round 1. These scores could be used to improve specific aspects of the systems for Round 2. The results of all teams on the specific types of matching challenges are found in Table 5. Team ISCAS-ICIP was able to improve on their performance for all challenges in Round 2, significantly improving on new products, which were

	Team			
Dimension	PMap	Rhinobird	ISCAS-ICIP	ASVinSpace
pre-processing	remove symbols and non-alphabet chars	remove stopwords and lower-case	remove stopwords, alphanumeric chars and lower-case	-
used attributes	title	title, description	title, price brand (extracted) model (extracted)	title, description, specTableContent
matching model	BERT-large RoBERTa-large RoBERTa-base	BERT-base	MPM HierMatcher Ditto	DistilRoBERTa-base
matching decision	ensemble of transformers	ensemble of self-ensembling transformers with different loss functions	ensemble of different model types	single model
post-processing	-	heuristic rule to correct predictions	heuristic rules to correct predictions	-
external resources	-	training data spanning four categories from WDC corpus	WDC corpus used for additional training data and to build vocabularies for attribute extraction	training data spanning four categories from WDC corpus

Table 4. Overview of the systems submitted for Task 1

not contained in the provided training set. For Round 2 they exchanged one of the matching models in their ensemble with a transformer-based model. This may suggest that this kind of model is better suited for handling new products than the previously used one (see below). Their overall result improved by 3% F1 going from Round 1 to 2. Team Rhinobird manages to improve significantly for products containing typos or dropped words while losing some performance across the other classes. They changed one of the BERT models in their ensemble of three to one trained with a different loss function for Round 2. Their overall result improves by only 0.5% F1 from Round 1 to 2, trading 2% Precision for 4% Recall. Overall, there is no team that consistently beats the others across all challenges, which is not surprising, as they all apply similar approaches and the overall results of the top teams vary only within 1% F1.

4.2 Results Task 2 - Product Classification

Table 6 shows the results of the participating systems on Task 2. Team Rhinobird obtained the highest overall WAF1 of 88.62, significantly higher than the baseline. Also, four out of five teams managed to outperform the baseline. Comparing the performance over the three different classification levels, understandably, Level-3 is the most difficult, due to a significantly larger labelset and consequently, fewer training examples per class.

Among teams that submitted their system description paper, in terms of supervised learning, all participants used Deep Neural Network structures and the more recent transformer-based architectures or pretrained language models. This aligns with the overall trend in the use of machine learning based methods for NLP, where we see a shift towards using pretrained language models often using transformer based architectures. The DNN models range from simple adaptation

	F1 on specific type of matching challenge				
Team	SN-DM	NP-HS	NP-LS	KP-TY	KP-DR
PMap	87.03	73.85	72.46	**87.01**	91.30
Rhinobird (Round 2)	87.56	68.49	71.43	85.06	**93.62**
Rhinobird (Round 1)	**88.04**	72.46	**76.92**	80.95	89.50
ISCAS-ICIP (Round 2)	87.97	**90.91**	67.69	74.21	91.30
ASVinSpace	87.41	82.14	76.19	75.78	84.39
ISCAS-ICIP (Round 1)	87.20	81.48	60.00	72.61	85.71
Megagon	80.00	71.19	72.73	47.33	71.79
Team ISI	82.52	24.00	23.26	27.59	63.01

Table 5. Results on the specific types of matching challenges (see Table 2)

of existing pre-trained language models (e.g., DICE) to very complex structures that combine 17 different models through ensemble (e.g., Rhinobird).

In terms of the text input used for supervised learning, all teams used product name and description, except ASVinSpace that also used URL. However, it is unclear what the effect of URL is, due to a lack of ablation test. Interesting, no teams used the category text provided as-is by the source vendor websites, even thought such content proves to be useful for such tasks [16, 22]

In terms of using external resources (excluding the use of pre-trained language models) to support the learning, Team ASVinSpace used a novel approach that extends the training set by harvesting data from Wikidata. None of the teams used the pre-trained embedding models or the product description corpus. However, Table 6 demonstrates that the pre-trained embedding models can be effective for further enhancing the learning.

Teams	Weighted Macro-F1 per level			Average over all levels		
	Lvl.1	Lvl.2	Lvl.3	Precision	Recall	WAF1
Rhinobird (Round 1)	91.83	90.11	84.68	89.01	89.04	88.62
Rhinobird (Round 2)	90.81	90.12	84.37	88.97	88.72	88.43
ISI	88.54	87.30	83.77	87.16	86.85	86.54
ASVinSpace	89.44	88.05	80.86	86.96	86.30	86.10
Megagon	88.23	85.47	81.23	85.39	85.38	84.98
DICE	85.79	81.46	78.27	85.30	81.49	81.84
Baseline	86.56	85.31	80.89	85.53	84.17	84.26
Baseline + CBow	88.06	86.97	82.17	86.50	86.07	85.73
Baseline + Skipgram	86.87	85.99	80.86	85.45	84.91	84.58

Table 6. Participants results on Task 2, product classification. WAF: Weighted Average (macro) F1. Baseline + CBow and Baseline + Skip indicates the results of the baseline when the product embedding models are used.

4.3 Participating Teams

In the following, we summarize the approaches that were used by the different teams. The complete details about the methods are given in the systems papers written by the teams themselves which are contained in the MWPD2020 proceedings.

Team Rhinobird represents the Tongji University of China and Tencent, China, and participated in both Task 1 and 2 in both rounds. For task 1, they rely on the BERT model while experimenting with different loss functions and ensembling steps. More specifically, after removing stopwords and lower-casing the data, they fine-tune multiple BERT models while experimenting with different training sets, features as well as focal loss [13] as a variation to the standard cross-entropy loss function. In addition to the provided training set, they experiment with a larger training set containing product pairs for four product categories. They try using only the *title* attribute as well as the concatentation of *title* and *description* as input features. Furthermore, Team Rhinobird experiments with a method of self-ensembling across multiple training epochs of the same model, namely stochastic weight averaging (SWA) [10]. Finally, subsets of all the previously mentioned models are ensembled by averaging their prediction probabilities and subsequently selecting the best performing ensemble. Team Rhinobird also applies some simple post-processing rules to correct predictions of the models, more specifically, all test pairs that do not belong to the same product category, are set to to be *non-matches*. Their submission for both rounds consists of an ensemble across three fine-tuned BERT models trained with different choices for the previously mentioned parameters.

For Task 2, Rhinobird used a BERT-based ensemble model that explicitly considers the dependencies among different category levels. Such hierarchical dependency features are encoded using a dynamic masked matrix obtained based on the hierarchical category structure. The masked matrix as a filter that dynamically discards the child categories irrelevant to the current parent category. The final ensemble model combines 17 different BERT models to make the final decisions. They used both product names and descriptions.

Team PMap represents the National Institute of Advanced Industrial Science and Technology, Japan, and participated in Task 1 in Round 1 only. They rely on using pretrained transformer-based language models, more specifically they fine-tune BERT-base, BERT-large, DistilBERT-base, RoBERTa-base and RoBERTa-large, consequently ensembling the results of some of these models to arrive at the final matching decision. Before fine-tuning they apply simple preprocessing, e.g. removing symbols and non-alphabet characters using a regular expression. Team PMap uses the datasets provided during the challenge without further additions. They furthermore use only the *title* attribute as input feature. After fine-tuning each model, based on their results, they select the BERT-large, RoBERTa-large and RoBERTa-base models for ensembling to reach the final matching decision.

Team ASVinSpace represents the Leipzig University, Germany and the German Aerospace Center (DLR). They participated in both Task 1 and 2 in

Round 1. For Task 1, they employed pre-trained transformer-based language models, namely BERT, RoBERTa and their distilled versions [21]. Different feature combinations are tried with the input to the model consisting of the concatenation of the used feature strings. The standard model is augmented with a single dense and an output layer on top of the pooled output of the [CLS] token and subsequently fine-tuned. ASVinSpace try solving the task in two ways, once minimizing cross-entropy loss on an output layer of size two and on the other hand framed as a regression problem with a single output and minimizing the mean squared error. In addition to the training set provided for the challenge, the team further experiments with additional training data from other product categories from the same data corpus. To handle the class imbalance inherent to the data, the team randomly drops negative examples during each training epoch to normalize the class distribution. The final submitted result is achieved by a DistilRoBERTa model using the *title*, *description* and *specTableContent* attributes, fine-tuned on data from four different product categories.

For Task 2, ASVinSpace used a CNN model adapted from [11]. It used a transformer-based language model [21] as input to the CNN layers instead of static word embedding models. The CNN model has three output layers, each corresponding to one of the classification levels, thus allowing the model to capture inter-dependencies of the different classification tasks. In addition, ASVinSpace also proposed to use external resources. For example, names of examples from the training set are used to retrieve relevant entities from Wikidata. Then, the corresponding descriptions and the GPC standard are used to disambiguate the retrieved entities in order to select only the ones that are highly similar (using a cosine similarity metric based on Tf-IDF weighted feature vectors) to the classification examples. These 'expanded' entities are manually annotated to create additional training data. In terms of the text input, they used the concatenation of product names, descriptions, and URLs.

Team ISCAS-ICIP represents the Chinese Academy of Sciences, and participated in Task 1 in Round 1 and 2. Their approach is based on three steps: pre-processing, entity matching and post-processing. During pre-processing the team removes stopwords, alphanumeric characters and lower-cases the examples. Furthermore, they apply a feature extraction approach based on vocabularies built using the provided data corpus as well as based on regex-patterns to extract values for the attributes brand and model. For the entity matching stage, the team applies overall four different models whose results are integrated to achieve the final prediction using a voting mechanism. The models are MPM [9], Seq2SeqMatcher [18], HierMatcher [8] and Ditto [12]. Finally, the post-processing module uses rules to correct predictions under certain circumstances, e.g. always assigning the label"*match* if two products have an exact match on the *title* attribute, or always assigning *non-match* if the brands of two products differ. The team also augments the provided training and validation sets by doubling their number using a similar sampling approach to the one used for building the provided training sets [20]. For their first round submission, ISCAS-ICIP integrated the results of the MPM, Seq2SeqMatcher and Hiermatcher models, while for the

second round submission they omit Seq2SeqMatcher and replace it with Ditto, which is based on pre-trained transformer-based language models, leading to improved performance (∼3% F1) on the evaluation test set.

Team DICE represents the Paderborn University, Germany, and participated in Task 2 in Round 1 only. The team used a simple adaptation of the BERT language model, by adding on top a fully-connected layer (i.e Dense) and using using a sigmoid activation function as replacement of the original softmax for classification. They used both product names and descriptions as input.

Team Megagon represents megagon.ai and **Team ISI** represents the University of Southern California, USA. They participated in both Tasks in Round 1, but did not submit a system paper.

5 Conclusion

The systems that were successful in the challenge all employed pre-trained transformer based language models which underlines the potential of these models for Web data integration tasks [4, 19, 12]. Especially the good results of systems using RoBERTa show the benefits of transferring knowledge that has been learned using less structured web content from diverse sources[12] to integration tasks involving structured web content, such as the matching and categorization tasks addressed in the challenge.

Several other benchmark competitions on product matching and product classification have been conducted in the last years: *The SIGIR 2018 eCom Rakuten Data Challenge* [14] focused on product classification, where individual products are classified into a hierarchy of over 3,000 categories in a company-specific catalogue (i.e., the Rakuten product catalogue). Compared to the Rakuten challenge which only involved product descriptions from a single source, our classification task involves more heterogeneous product descriptions from many websites. The 2019 and 2020 workshops on *Challenges and Experiences from Data Integration to Knowledge Graphs* (DI2KG2019) [7][13] and (DI2KG2020)[14] focus on knowledge graph creation from product specifications which were extracted from the Web. The workshops feature three shared tasks: entity resolution, schema matching, attribute matching. Products are described in the DI2KG dataset using distinct attributes such as screen size, display type, or refresh rate. Compared to the DI2KG entity resolution task, our matching task involves dealing with less structured textual product data.

Based on the findings from this event, we identify several remaining gaps in the current research: First, despite the dominance of transformer based language models, there remains a significant degree of variety in terms of how such models can be adapted and/or combined for data integration tasks. There is also a lack

[12] RoBERTa was pre-trained using different subsets of the CommonCrawl along with several text corpora.
[13] http://di2kg.inf.uniroma3.it/2019/
[14] http://di2kg.inf.uniroma3.it/2020/

of systematic study of how these architectures compare under a uniform experimental setting. Second, there is a lack of exploration into what kind of external resources can be used to support such tasks and how they can be used to do so. For example, our product data textual corpus could be used to fine-tune a language model following an approach such as [2], which showed further gain in terms of domain-specific tasks [19]. Finally, in terms of mining product information on the Web from a more general point of view, recent research [1] focused on harvesting and cleaning structured product data on the Web. However, there is a lack of studies on how such data could be used to enable self-supervised learning in downstream tasks [4]. We encourage future research to investigate these directions.

Acknowledgements This event is partially sponsored by Peak Indicators, UK[15].

References

1. Bai, P., Ge, Y., Liu, F., Lu, H.: Joint interaction with context operation for collaborative filtering. Pattern Recognition **88**, 729–738 (2019). https://doi.org/doi:10.1016/j.patcog.2018.12.003
2. Beltagy, I., Lo, K., Cohan, A.: Scibert: A pretrained language model for scientific text. arXiv preprint arXiv:1903.10676 (2019)
3. Bizer, C., Primpeli, A., Peeters, R.: Using the semantic web as a source of training data. Datenbank-Spektrum **19**(2), 127–135 (2019). https://doi.org/10.1007/s13222-019-00313-y
4. Deng, X., Sun, H., Lees, A., Wu, Y., Yu, C.: TURL: Table Understanding through Representation Learning. arXiv:2006.14806 [cs] (Jun 2020)
5. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 4171–4186 (Jun 2019)
6. Dong, X.L.: Challenges and innovations in building a product knowledge graph. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. p. 2869. Association for Computing Machinery (2018)
7. Firmani, D., Crescenzi, V., Angelis, A.D., Dong, X.L., Mazzei, M., Merialdo, P., Srivastava, D.: Proceedings of the 1st international workshop on challenges and experiences from data integration to knowledge graphs. CEUR Workshop Proceedings, vol. 2512. CEUR-WS.org (2019)
8. Fu, C., Han, X., He, J., Sun, L.: Hierarchical Matching Network for Heterogeneous Entity Resolution. In: Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence. pp. 3665–3671 (Jul 2020)
9. Fu, C., Han, X., Sun, L., Chen, B., Zhang, W., et al.: End-to-End Multi-Perspective Matching for Entity Resolution. In: Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence. pp. 4961–4967 (2019)
10. Izmailov, P., Podoprikhin, D., Garipov, T., Vetrov, D., Wilson, A.G.: Averaging Weights Leads to Wider Optima and Better Generalization. arXiv:1803.05407 [cs, stat] (Feb 2019)

[15] https://www.peakindicators.com/

11. Kim, Y.: Convolutional neural networks for sentence classification. In: Proc. of the Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 1746–1751. Association for Computational Linguistics (ACL) (2014)
12. Li, Y., Li, J., Suhara, Y., Doan, A., Tan, W.C.: Deep Entity Matching with Pre-Trained Language Models. arXiv:2004.00584 [cs] (Apr 2020)
13. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollar, P.: Focal Loss for Dense Object Detection. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2980–2988 (2017)
14. Lin, Y.C., Das, P., Datta, A.: Overview of the sigir 2018 ecom rakuten data challenge. In: eCOM at The 41st International ACM SIGIR Conference on Research and Development in Information Retrieval. CEUR-WS.org (2018)
15. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., et al.: RoBERTa: A Robustly Optimized BERT Pretraining Approach. arXiv:1907.11692 (2019)
16. Meusel, R., Primpeli, A., Meilicke, C., Paulheim, H., Bizer, C.: Exploiting microdata annotations to consistently categorize product offers at web scale. In: International Conference on Electronic Commerce and Web Technologies. pp. 83–99. Springer International Publishing (2015)
17. Mudgal, S., Li, H., Rekatsinas, T., Doan, A., Park, Y., et al.: Deep Learning for Entity Matching: A Design Space Exploration. In: Proceedings of the 2018 International Conference on Management of Data. pp. 19–34 (2018)
18. Nie, H., Han, X., He, B., Sun, L., Chen, B., et al.: Deep Sequence-to-Sequence Entity Matching for Heterogeneous Entity Resolution. In: Proceedings of the 28th ACM International Conference on Information and Knowledge Management. pp. 629–638 (Nov 2019)
19. Peeters, R., Bizer, C., Glavaš, G.: Intermediate Training of BERT for Product Matching. In: DI2KG Workshop @ VLDB (2020)
20. Primpeli, A., Peeters, R., Bizer, C.: The WDC Training Dataset and Gold Standard for Large-Scale Product Matching. In: Workshop on e-Commerce and NLP (ECNLP2019), Companion Proceedings of WWW. pp. 381–386 (2019)
21. Sanh, V., Debut, L., Chaumond, J., Wolf, T.: Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. In: NeurIPS EMC2 Workshop (2019)
22. Zhang, Z., Paramita, M.: Product classification using microdata annotations. In: Ghidini, C., Hartig, O., Maleshkova, M., Svátek, V., Cruz, I., Hogan, A., Song, J., Lefrançois, M., Gandon, F. (eds.) The Semantic Web – ISWC 2019. pp. 716–732. Springer International Publishing (2019)