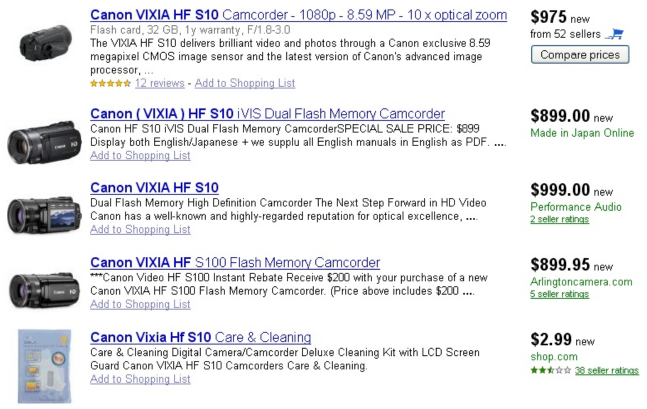
Tailoring entity resolution for matching product offers

Hanna Köpcke Andreas Thor Stefan Thomas Erhard Rahm
Web Data Integration Lab, Leipzig, Germany
{koepcke, thor, rahm}@informatik.uni-leipzig.de, mam07bmm@studserv.uni-leipzig.de

ABSTRACT

Product matching is a challenging variation of entity resolution to identify representations and offers referring to the same product. Product matching is highly difficult due to the broad spectrum of products, many similar but different products, frequently missing or wrong values, and the textual nature of product titles and descriptions. We propose the use of tailored approaches for product matching based on a preprocessing of product offers to extract and clean new attributes usable for matching. In particular, we propose a new approach to extract and use so-called product codes to identify products and distinguish them from similar product variations. We evaluate the effectiveness of the proposed approaches with challenging real-life datasets with product offers from online shops. We also show that the UPC information in product offers is often error-prone and can lead to insufficient match decisions.


Canon VIXIA HF S10 Camcorder - 1080p - 8.59 MP - 10 x optical zoom
Flash card, 32 GB, 1y warranty, F/1.8-3.0
The VIXIA HF S10 delivers brilliant video and photos through a Canon exclusive 8.59 megapixel CMOS image sensor and the latest version of Canon's advanced image processor, ...
12 reviews - Add to Shopping List
$975 new
from 52 sellers
Compare prices

Canon (VIXIA) HF S10 iVIS Dual Flash Memory Camcorder
Canon HF S10 iVIS Dual Flash Memory CamcorderSPECIAL SALE PRICE: $899 Display both English/Japanese + we supplu all English manuals in English as PDF. ...
Add to Shopping List
$899.00 new
Made in Japan Online

Canon VIXIA HF S10
Dual Flash Memory High Definition Camcorder The Next Step Forward in HD Video Canon has a well-known and highly-regarded reputation for optical excellence, ...
Add to Shopping List
$999.00 new
Performance Audio
2 seller ratings

Canon VIXIA HF S100 Flash Memory Camcorder
***Canon Video HF S100 Instant Rebate Receive $200 with your purchase of a new Canon VIXIA HF S100 Flash Memory Camcorder. (Price above includes $200 ...
Add to Shopping List
$899.95 new
Arlingtoncamera.com
5 seller ratings

Canon Vixia Hf S10 Care & Cleaning
Care & Cleaning Digital Camera/Camcorder Deluxe Cleaning Kit with LCD Screen Guard Canon VIXIA HF S10 Camcorders Care & Cleaning.
Add to Shopping List
$2.99 new
shop.com
38 seller ratings

Figure 1: Product offers related to the Canon VIXIA HF camcorder in Google Product Search

1. INTRODUCTION

Product matching deals with the identification of different descriptions or offers referring to the same real-world product. Given that many thousands of online shops sell millions of diverse products over the web, product matching has become of increasing importance. For example, it is a critical task for aggregating offers for the same product within price comparison portals (e.g., PriceGrabber), online marketplaces (e.g., Amazon.com), or product search engines (e.g., Google Product Search) [5][10].

Product matching is a special case of entity resolution (matching) that is needed to identify equivalent entities or duplicates within a data source or between data sources. While this problem has received a huge amount of effort in research (see [3, 7] for recent surveys), only little work has been devoted to product matching. Product matching for e-commerce websites introduces several specific challenges that make this problem much harder than other forms of entity resolution, e.g., to match records about publications. In particular, there is a huge degree of heterogeneity since product offers come from thousands of merchants using different names and descriptions of the products. Furthermore, offers frequently have missing or wrong values and are mostly not well structured but mix different product characteristics in text fields such as product name or description [5].

For illustration, we show in Figure 1 some result offers for the product search engine Google Product Search and a specific camcorder. The offers refer to different merchants that use heterogeneous names, descriptions, and other attributes for the same product and may also contain misspellings and other errors. For example, the product names for the considered product Canon Vixia HF S10 partially include specific technical details that may complicate product matching, e.g., to find out that (only) the first three entries refer to the same product. Note that Google already performs a product matching since the first entry refers to offers from 52 merchants. The remaining duplicates in the example show that Google's clustering is imperfect and needs to be improved.

A recent benchmark study [8] evaluated current entity matching prototypes and a commercial tool on different real-world match tasks including e-commerce product matching. Despite the use of small-sized datasets, current solutions could achieve only about 30 - 70% F-measure for product matching (compared to up to 98% F-measure for publication matching) underlining the high difficulty of product matching. In [5] a machine learning approach is presented to match product offers to comprehensive product descriptions. Their eval-

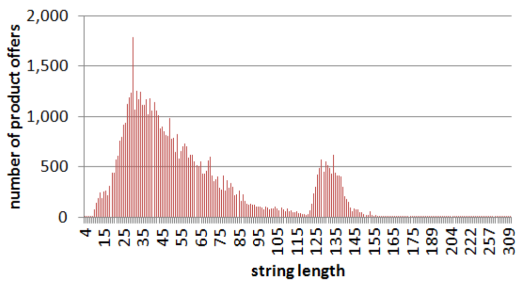


Figure 2: Distribution of string length for the product title attribute

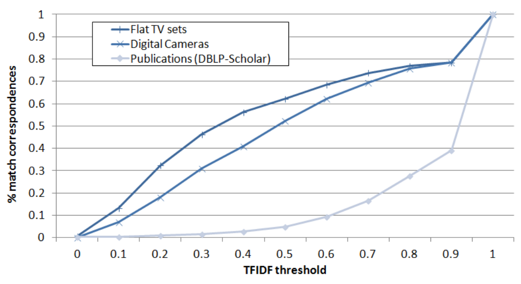


Figure 3: Cumulative distribution of TF/IDF similarity for match correspondences

uation focused on favorable categories for which a precision of at least 85% could be achieved; for the remaining categories they report that a TF/IDF-based matching was limited to about 40% F-measure on average.

In practice, product search engines and price comparison portals frequently rely on presumably unique product identifiers such as UPC (Universal Product Code) or GTIN (Global Trade Item Number) to match product offers and products. However such identifiers are frequently not available so that there is a need for additional product matching techniques. Furthermore, as we will show UPC usage in current offers is also error-prone and may lead to sub-optimal match results.

To improve product matching we advocate for the application of an extensive preprocessing to extract significant attribute values from textual attributes such as product names or descriptions. In particular, we propose the detection and use of so-called *product codes* that that frequently allow the identication of products of a certain type, e.g., electronic products. For our example in Figure 1, the manufacturer-specific product code HF S10 is common to the first three product offers. The example also illustrates some of the difficulties with product codes since small product code variation (such as for the fourth offer) can already refer to a different product. Furthermore, as shown by the last offer, product codes that appear within accessory product offers can be misleading since they may not identify the accessory product but the product for which the accessory can be used.

We evaluate our product matching approach for a large real-life dataset containing more than 100,000 online offers for electronic products and accessories of many categories. Figure 2 shows the length distribution of the product titles for these offers. It illustrates that this important attribute is highly heterogeneous with many long, verbose strings. The puzzling second peak around string length 135 is due to numerous accessory products for digital cameras, mobile phones, and navigation systems whose titles contain a long list of models for which they are suitable. As a result, standard string measures are likely of limited effectiveness. This is further confirmed by Figure 3 showing for two product match tasks and a publication match task (from [8]) the percentage of correspondences (matching entity pairs) with a string (TF/IDF) similarity smaller or equal to a given similarity threshold t. For example, approx. 60% of all correspondences of the product category Digital Cameras have a title similarity below or equal to 0.5 making it difficult to identify matching offers. On the other hand, matching publications is much easier since 60% of the correspondences have a title similarity of more than 90%.

In the next section, we outline our overall approach to matching product offers. It supports category-specific match strategies and is based on machine learning to semi-automatically determine a match strategy utilizing several attributes and similarity functions. Our new approach to extract and verify product codes is described in Section 3. We then use our real-life dataset to evaluate the effectiveness of the extraction approach and its utility for matching offers for electronic products and associated accessory products. To evaluate the match quality we compare against two reference mappings, one based on UPC values and one manually determined "perfect" mapping. This evaluation also reveals limitations of UPC-based mappings and thus on using UPC values for product matching.

2. SYSTEM DESIGN

Given a collection of product offers our goal is to identify corresponding offers, i.e., offers that refer to the same real-world product. Typically only few attribute values are available per offer, in particular product title, product description, manufacturer, price, and perhaps a product identification such as UPC. While we consider the UPC values for evaluation, we are only concerned here with matching based on the remaining information. Matching product offers against other offers is much more challenging than matching product offers to an existing catalog of consolidated product descriptions (e.g., as done in [5]) due to the absence of structured and cleaned product data. Directly matching product offers is highly relevant since in many scenarios a carefully maintained reference product catalog is not available. For example, price monitoring applications that keep track of offers from different merchants or websites typically start without a product catalog. Offer matching is also useful to aggregate product information for an incremental construction of product catalogs.

Matching noisy real-world entities such as product offers requires the combined use of several matchers (e.g., different string similarity measures [2]) on several attributes to derive a match decision for every pair of entities. Given the availability of several relevant attributes and similarity measures, it becomes a very complex task to manually specify a reasonable strategy for the combination of matcher similarities.

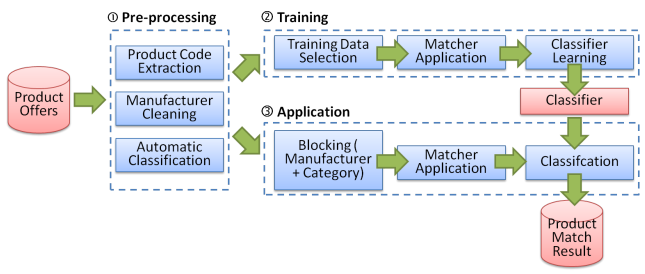


Figure 4: Overall workflow for matching product offers

Therefore, we employ a learning-based approach and treat product matching as a classification problem.

The overall design of our system is illustrated in Figure 4. The match workflow runs in three phases: pre-processing, training, and model application. The **product code extraction** is one element of preprocessing and will be described in Section 3 in more detail. Further preprocessing tasks are the cleaning of manufacturer information and the categorization of product offers.

For **manufacturer name cleaning** we cluster different variations of the same manufacturer based on a combination of string similarities, synonym lists, and existing lists of manufacturers. Since a significant number of product offers does not provide a manufacturer attribute value, we also analyze the product title and descriptions for manufacturer names from the manufacturer dictionary. Space restrictions prevent us from providing more details on manufacturer cleaning.

An important use case for product matching that we consider is the assignment of product offers to a given set of product categories. Such a product **categorization** [1] allows the restriction of matching to offers from the same category thereby improving match efficiency and likely also match precision. Furthermore, it is possible to devise category-specific match strategies to take characteristics of certain product types into account. We assign offers to one of the categories by using a Modified Naïve Bayes approach according to [6] and utilizing already assigned offers. To classify an offer o, the Naïve Bayes Classifier calculates the posterior probability $P(c|o)$ for all categories c. To this end, attribute values (e.g., title, description, manufacturer) of already categorized offers are tokenized and the probabilities $P(t|c)$ are computed for all tokens t. Tokens and their probabilities are weighted based on their attribute. Again we are unable to provide more details due to space restrictions.

The training phase requires **selection of training data**, i.e., entity pairs that are annotated whether they represent a match or a non-match. The manual labeling of training data is very time consuming and is typically done offline. In our approach match and non-match pairs are generated utilizing the ratio approach as described in [9]. This strategy considers only entity pairs for labeling, for which the similarity exceeds a specified threshold. This ensures that the training is not dominated by trivial non-matching entity pairs that are not useful for finding effective matcher

Algorithm 1: Product code extraction

```
getProductCode(offer, regularExpressions, threshold)
    // remove common features, e.g., dimensions, weight, ...
    offer ← removeFeatures(offer);
    // tokenize and remove stop words and frequent
    // category-specific tokens
    tokens ← tokenize(offer);
    tokens ← removeFrequentTokens(tokens);
    // candidates (= up to 3 tokens) that satisfy
    // specific patterns
    candidates ← generateCandidates(tokens);
    candidates ← filterCandidates(regularExpressions);
    // get code with highest web score above threshold
    code ← webVerification(candidates, threshold);
    return code;
```

parameters and combinations. Furthermore it aims at a certain ratio of matching and non-matching entity pairs in the training data to enhance the training data's discriminative value for learning effective match strategies.

As the characteristics of product data and the variety and distribution of errors across different categories can vary greatly we adopt an **adaptive learning approach** for training separate classifiers for each category. To this end we apply the learning individually for each category using disjoint subsets of training data according to the categories of the labeled training data. We also support learning a uniform model (match strategy) across all categories. In the evaluation we will compare the effectiveness of the uniform approach with category-specific match strategies. Learning is performed for several matchers, in particular for three string measures (TF/IDF, Trigram Jaccard) on the title and description attributes and a specific product code matcher (see Section 3). For the learner we decided on the Support Vector Machine (SVM). The SVM has already been investigated for entity resolution in several previous studies and proven to produce stable results.

During the last phase, **application**, the learned match strategies are applied to determine matching product offers. To reduce the search space we use the cleaned manufacturer value and the product category for blocking, i.e., we apply matchers only for pairs of product offers sharing the same manufacturer and category. The resulting match similarities are the input for the learned classification model that, in turn, provides the "match or non-match" decision.

3. PRODUCT CODE EXTRACTION

One key observation is the frequent existence of specific product codes for certain product types that can help to differentiate similar but different products. A product code is a manufacturer-specific identifier that typically appears in the product title and description. In general, it can be any sequence consisting of alphabetic, special, and numeric characters split by an arbitrary number of white spaces. In the example of Figure 1 the term HF S10 is a product code for the first three entries. A product code is under full control of the manufacturer and thus we observe very good data quality, i.e., the product code is usually correct if it is available. Unfortunately, product codes are generally not provided as a separate attribute but appear only within the product title or description.

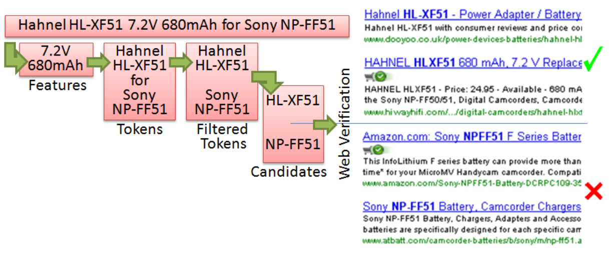

Hahnel HL-XF51 7.2V 680mAh for Sony NP-FF51
7.2V
680mAh
Features
Hahnel
HL-XF51
for
Sony
NP-FF51
Tokens
Hahnel
HL-XF51
Sony
NP-FF51
Filtered
Tokens
HL-XF51
NP-FF51
Candidates
Web Verification
Hahnel HL-XF51 - Power Adapter / Battery
Hahnel HL-XF51 with consumer reviews and price co
www.dooyoo.co.uk/power-devices-batteries/hahnel-hl
HAHNEL HLXF51 680 mAh, 7.2 V Replace
HAHNEL HLXF51 - Price: 24.95 - Available - 680 mA
the Sony NP-FF50/51, Digital Camcorders, Camcorde
www.hiwayhifi.com/.../digital-camcorders/hahnel-hlx
Amazon.com: Sony NPFF51 F Series Batter
This InfoLithium F series battery can provide more than
time* for your MicroMV Handycam camcorder. Compati
www.amazon.com/Sony-NPFF51-Battery-DCRPC109-35
Sony NP-FF51 Battery, Camcorder Chargers
Sony NP-FF51 Battery, Chargers, Adapters and Accesso
batteries are specifically designed for each specific cam
www.atbatt.com/camcorder-batteries/b/sony/m/np-ff51.a

Figure 5: Example code extraction for Hahnel HL-XF51 7.2V 680mAh for Sony NP-FF51 **(manufacturer:** Hahnel**).**

The extraction of the product code of the offered product is non-trivial as the title and the description of the product offer contain several unstructured information. Furthermore, accessory products may also contain multiple product codes, e.g., one for the accessory itself and one for the target product. Product code extraction is a special case of product attribute extraction that identifies attribute-value pairs out of unstructured textual descriptions (e.g., [4]). However, such approaches typically require labeled (tagged) training data whereas our focused product code extraction does not need any training data but employs the rich knowledge of search engines. The product code extraction algorithm is illustrated in Algorithm 1 and will be described next. For illustration purposes Figure 5 demonstrates the extraction workflow for the sample product title Hahnel HL-XF51 7.2V 680mAh for Sony NP-FF51.

The first step, **feature extraction**, applies regular expressions to extract common features such as dimensions, weight specification, colors, etc. In our example the voltage (7.2V) and energy (680mAh) are extracted. The next step, **tokenization**, breaks the title string into words. Tokens are separated by white spaces and punctuations.

Filtering comprises the removal of stop words as well as other tokens that appear frequently in product offers of several different manufacturers. For this we calculate a manufacturer-based frequency for each token t appearing in any offer representation. Let $N(t, m)$ be the number of product offers of manufacturer m containing the token t and let $N(t)$ be the overall number of product offers containing t. For any product offer of m only tokens t with a ratio $N(t, m)/N(t)$ above a given threshold are considered for product code extraction. In our experiments we employ a threshold of 50%, i.e., at least 50% of all product offers containing t must be from manufacturer m. In the running example the term for will thus be excluded from further steps.

Afterwards we **generate candidates** for product codes. In general, a candidate consists of up to 3 consecutive tokens. To reduce the possibly large number of candidates regular expressions are employed to find "interesting" candidates, e.g., candidates that contain both letters and numbers. To this end we use a manually created list of regular expressions that captures knowledge on the syntactical structure of common product codes. Furthermore, string type frequencies can be computed to identify types that frequently occur with a particular manufacturer. For example, a significant number of candidates for the manufacturer Hahnel follows

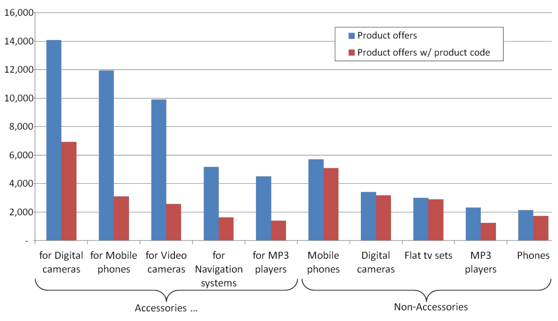


Figure 6: Number of product offers for 10 product categories (5 accessory and 5 non-accessory categories).

the pattern [A-Z]{2}\-[A-Z]{2}[0-9]{2} ("two capital letters, minus, two capital letters, two digits") which indicates that such strings can be product codes.

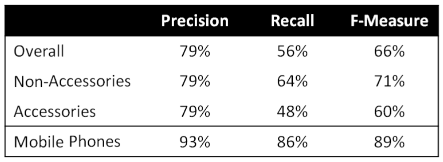
Finally, a **web verification** step utilizes the web as an external knowledge source to verify the extracted candidates. For each of the determined candidates a query is submitted to a web search engine. The correctness of a code candidate is verified by the ratio of the results containing the corresponding manufacturer. Figure 5 illustrates for the two candidates HL-XF51 and NP-FF51 the retrieved top 2 query results. For the first candidate, HL-XF51, all results contain the manufacturer name Hahnel and thus giving an overlap of 100%. The term HL-XF51 is therefore considered a valid product code. On the other hand, NP-FF51 is *not* a product code because none of the results contain the manufacturer name Hahnel.

4. EVALUATION

To evaluate our product matching approaches we use a large real-world dataset provided by an e-commerce portal [1]. The evaluation dataset comprises a total of 102,182 offers for electronic products and accessory products that are associated to 71 given product categories of the portal. The offers are mostly limited to only a few attributes, in particular product title, description, and manufacturer. For evaluation purposes, we also required that all offers contain an UPC value assuming that it permits the evaluation of the quality (recall, precision, F-measure) of different match strategies. In addition, we manually determined a perfect match mapping for two selected product categories. All product offers are preprocessed as described, i.e., we cleaned the manufacturer values, extracted product codes if possible, and automatically assigned offers to product categories. Since the evaluation of manufacturer cleaning and product categorization is beyond the scope of this paper, we corrected the results of these steps manually to avoid negative side effects.

In the following, we first evaluate the proposed approach to derive product codes from the titles of product offers of different categories. We then evaluate the effectiveness of different general and category-specific strategies for matching product offers and study the usefulness of using the extracted product codes. This evaluation is first done w.r.t.

[1]Because of a non-disclosure agreement we are not able to provide any details on the e-commerce portal.



	Precision	Recall	F-Measure
Overall	79%	56%	66%
Non-Accessories	79%	64%	71%
Accessories	79%	48%	60%
Mobile Phones	93%	86%	89%

Figure 7: Quality of product code extraction

an UPC-based reference mapping assuming that only offers with the same UPC are matching. We then study the match quality w.r.t. the manually determined perfect mapping and discuss limitations of relying on UPC values only.

4.1 Extraction of product codes

The effectiveness of the proposed approach to extract product codes depends on the product category, especially on whether the offers refer to accessory products or to the main, non-accessory products. In Figure 6 we show the number of product offers for the largest five accessory categories as well as the largest five non-accessory categories. Furthermore, we show for each category for how many offers a product code could be extracted by our approach. We observe significant differences between accessory and non-accessory products. For non-accessory product offers, we could mostly find a product code (in 85%, on average) in particular for the larger categories of mobile phones, cameras, and tv sets. By contrast, for the accessory product offers (which are much more frequent than non-accessory offers) a product code could be determined only in 34% on average.

To determine the quality of the extracted product codes we manually determined the correct product codes for a random subset of about 2% of the product offers over all categories. Figure 7 shows the quality results for product code extraction after the web-based verification. This verification proved to be an important step that helped to improve the results by up to 20% even for a low threshold value of 0.1. As shown, the average precision of the product codes is 79% over all categories, i.e., that almost four fifth of the found product codes are correct. The achieved recall values are smaller, especially for accessory products where less than half (48%) of the product codes could be found. The resulting F-measure is thus higher for non-accessory offers than for accessories. For mobile phones, product code extraction was most effective with an F-measure of 89%. The results show that product code extraction works best for non-accessory products so that they are likely to benefit more than accessory products from using them for matching product offers.

4.2 Match quality against UPC mapping

We evaluate the match quality for product offers of the five largest accessory and five largest non-accessory categories. As described in Section 2, we apply SVM-based match strategies combining different string measures on the product title and product description. Optionally, we consider an additional product code matcher requiring equal product codes for product offers to match. For training we use 1,000 matching and non-matching offer pairs per category. The training data is used to determine both a common match strategy for all categories as well as category-specific match strategies.

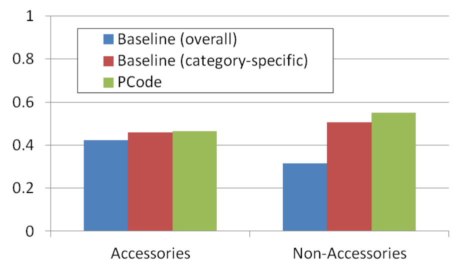


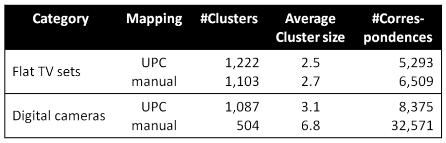
Figure 8: Quality of baseline and product code matching

We first evaluate match quality against an UPC-based reference mapping where only offers with the same UPC are considered to match (recall that we only evaluate offers for which the UPC is provided). Figure 8 shows the resulting average F-measure results over the considered accessory and non-accessory categories for three match strategies: a common baseline strategy (not using product codes) for all categories, category-specific baseline match strategies, and category-specific match strategies including the product code matcher. We first observe that the average F-measure values are generally rather low which is due to the high difficulty of matching offers against offers (and not against cleaned product descriptions). Furthermore, as we will see in the next subsection the comparison against the UPC reference mapping leads to pessimistic results.

We find that the use of category-specific match strategies generally outperforms the use of a common match strategy. This effect is especially pronounced for non-accessory product offers for which the average F-measure becomes higher than for accessory products. Similarly, the use of the product code matcher is most beneficial for non-accessory offers influenced by the improved coverage and quality of product code extraction as discussed above. Product code matching helps improve the average F-measure for non-acessory categories to 55%. The best match quality (F-measure 73%) is achieved for offers of mobile phones.

4.3 Match quality against manually determined reference mapping

A closer inspection of the UPC values for the product offers revealed several anomalies that questioned the appropriateness of UPC-based match decisions. There are offers for the same or almost the same product that come with different UPC values. One of the reasons for this phenomenon is that products may come with different codes based on the manufacturer's country or target market. For example, there are three different UPCs for the Canon IXUS 90 IS camera in our dataset. Furthermore, we observed the existence of offers for different products having the same UPC. So we decided to manually determine a reference match mapping to evaluate the quality of the automatic match strategies. For this purpose, we employed a crowd-sourcing effort involving several local researchers. Due to the large number of offers we still had to restrict the determination of the reference match mapping to two categories (flat tv sets and digital cameras). The manual match decisions did not try to differentiate all minor variations of the same product (e.g.,



Category	Mapping	#Clusters	Average Cluster size	#Corres-pondences
Flat TV sets	UPC	1,222	2.5	5,293
	manual	1,103	2.7	6,509
Digital cameras	UPC	1,087	3.1	8,375
	manual	504	6.8	32,571

Figure 9: Reference mappings

different colors) but had in mind what should be bundled in a comparison portal or product search engine to allow a meaningful price comparison, to utilize aggregated product reviews etc. Similar as in the example of Figure 1, the product code information turned out to be a useful indicator for match decisions.

Figure 9 provides some base statistics about the manually determined and the UPC-based reference mappings. The number of clusters indicates how many different products are distinguished per category, the cluster size indicates the average number of offers for the same product, and the number of correspondences specifies the number of matching pairs of offers. We observe that the manual mappings contain more correspondences than the UPC mappings by considering more offers to refer to the same product. The differences are especially pronounced for the camera category where the average number of offers per product more than doubles, i.e., the offers in most correspondences had different UPC values. To a much smaller extent, we observed that the same UPC value was assigned to non-matching offers with different product codes (for 90 correspondences of TV set offers and 331 pairs of camera offers).

Figure 10 compares the match quality (F-measure) for the two product categories w.r.t. both reference mappings. We only consider category-specific match strategies. For the UPC-based reference mapping, the baseline match strategy (not using product codes) performs similarly for both categories. For the tv set category, product code matching helped to substantially improve F-measure already for the UPC reference mapping. Comparing against the manually determined reference mapping results in a further albeit relatively small improvement (to 69% F-measure) since the two reference mappings are relatively similar for this category. By contrast, for the camera category the UPC mapping did not enable taking much advantage from product code matching since many offers with the same product code had different UPC values. Here, using the manual reference mapping helped to almost double the F-measure result compared to the UPC mapping (to 81%) indicating the high value of product code matching.

While we could only evaluate the offers for two categories we expect similar trends for other categories. We conclude that UPC-based match evaluations tend to be too pessimistic and that UPC-based matching may leave many matching or comparable offers unmatched. The manual reference mappings also showed the high potential of the proposed product code matching.

5. CONCLUSIONS

Matching product offers is a challenging problem requiring sophisticated and tailored entity resolution approaches. We outlined and evaluated such an approach that is based on machine learning and a comprehensive preprocessing. In particular, we proposed a new approach for improving product matching based on a pattern-based extraction and web-based verification of so-called product codes. Our evaluation with a large real-life dataset showed the high benefit of product code matching, especially for non-accessory products. Furthermore, we found that category-specific match strategies should be applied. We also analyzed the use of UPC values for evaluating match strategies and for product matching and observed significant limitations. In particular, UPC-based match evaluations tend to be too pessimistic and UPC-based matching may leave many matching or comparable offers unmatched. In future work we will investigate techniques to further improve product matching and to utilize product codes for matching offers to products.

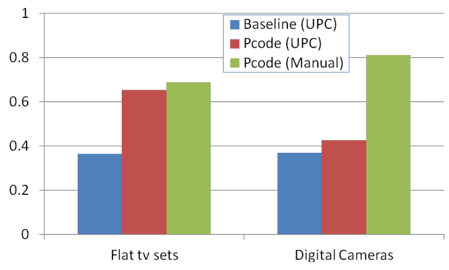


Figure 10: Match quality for different reference mappings

6. REFERENCES

[1] S. Bergamaschi, F. Guerra, and M. Vincini. A data integration framework for e-commerce product classification. *ISWC*, 2002.

[2] W. W. Cohen, P. D. Ravikumar, and S. E. Fienberg. A comparison of string distance metrics for name-matching tasks. In *IIWeb*, 2003.

[3] A. K. Elmagarmid, P. G. Ipeirotis, and V. S. Verykios. Duplicate record detection: A survey. *IEEE Trans. Knowl. Data Eng.*, 19(1), 2007.

[4] R. Ghani, K. Probst, Y. Liu, M. Krema, and A. Fano. Text mining for product attribute extraction. *ACM SIGKDD Explorations Newsletter*, 8(1):41–48, 2006.

[5] A. Kannan, I. E. Givoni, R. Agrawal, and A. Fuxman. Matching unstructured product offers to structured product specifications. In *Proc. KDD Conf.*, 2011.

[6] Y. Kim, T. Lee, J. Chun, and S. Lee. Modified Naïve Bayes Classifier for E-Catalog Classification. *Data Engineering Issues in E-Commerce and Services*, 2006.

[7] H. Köpcke and E. Rahm. Frameworks for entity matching: A comparison. *Data Knowl. Eng.*, 69(2), 2010.

[8] H. Köpcke, A. Thor, and E. Rahm. Evaluation of entity resolution approaches on real-world match problems. *PVLDB*, 3(1), 2010.

[9] H. Köpcke, A. Thor, and E. Rahm. Learning-based approaches for matching web data entities. *IEEE Internet Computing*, 99, 2010.

[10] H. Nguyen, A. Fuxman, S. Paparizos, J. Freire, and R. Agrawal. Synthesizing products for online catalogs. *PVLDB*, 4(7), 2011.