On the Investigation of Similarity Measures for Product Resolution

Krisztian Balog
Norwegian University of Science and Technology
Department of Computer and Information Science
Trondheim, Norway
krisztian.balog@idi.ntnu.no

Abstract

Entity resolution is an important information integration problem that has been looked at by a number of research communities. Resolution of a specific type of entity, products, however, is a largely unexplored area. This work sets out a series of first steps in that direction. We devise similarity measures for comparing products based on specific attributes. To estimate the discriminative power of these similarity functions we introduce a metric termed pairwise discriminative power. We report on experimental results using two purpose-built test sets, corresponding to two very different e-commerce segments: electronics and toys.

1 Introduction

E-commerce has become undeniably popular over the past decade. Almost any product imaginable can be purchased online: books, computers, clothing, furniture, flowers, and the list continues endlessly. The increase of e-commerce activity has had, and continues to have, a direct effect on the search industry. Product search systems are becoming indispensable tools for aiding users in making their selection decisions and finding the best offers in the ever-growing landscape of online web-shops. There are two main manifestations of product search systems: (i) verticals of major web search engines (such as Google product search[1] and Yahoo! shopping[2]) and (ii) price comparison websites (such as pricegrabber[3]). Both rely on information that retailers are required to make available: either as semantic markup on unstructured HTML documents (microdata, microformats, or RDFa) or as a data feed provided in some predefined structured format (e.g, XML or CSV) on a regular basis. Also of note that most price comparison sites use their own data schema; with the advent of microformats it is likely to change over time—but, even then, challenges remain.

Chief of these challenges is the resolution of products; webpages that represent the same product should be recognized. However, unique identifiers that could be used to join records (like EAN or SKU) are often absent or incompatible (as different sources use disparate means of identifying products). Further, the records representing the same product may have differing structure; one source might put the brand-name and the main product properties into a single name field (e.g,. *Apple iPod classic 160GB black*), while another organizes the same information into separate attributes: manufacturer, model, colour, and so on. The task of resolving entities has been looked at by a number of research communities, yet, we are not aware of any prior work focusing on the resolution of products in heterogeneous environments. This work sets out a series of first steps in that direction. Our main research objective is to devise measures for comparing products; first, on a per-attribute basis. These attribute-specific similarity functions can then be used as building blocks in more complex information integration scenarios.

Since no publicly available dataset exists for this purpose, we start off by constructing a test collection. As most prior research related to products focused on electronics (digital cameras, most prominently) we feel strongly about having test instances of another kind too. We build two test sets corresponding to two e-commerce segments that are of a very different nature: electronics and toys. The comparison of these two domains is woven throughout the whole paper. Product pages returned by a web search engine in response to an ad-hoc product query are used as candidate sets. Within each such set, products representing the same real-world entity are grouped manually into equivalence classes. Although there exist solutions for the automatic extraction of attributes from product pages, in the lack of training material, and, in the interest of avoiding any side-effects of using (potentially) erroneous data, we extracted product features manually.

To be able to measure how well similarity functions can can set apart entities within equivalence classes from other entities in the candidate set, we introduce a simple and intuitive metric termed *pairwise discriminative power* (PDP). This measure allows for the comparison of various similarity functions, even cross-domain, and also in cases where not all pairwise similarities can be computed, because of missing attributes. Finally, we introduce specific similarity functions for four main product features: name, price, manufacturer, and productID, and evaluate them using the proposed PDP measure. We find that the relative ordering of these four attributes is the same across the two domains.

[1] http://www.google.com/products

[2] http://shopping.yahoo.com

[3] http://www.pricegrabber.com

Our specific contributions are as follows: (1) we identify features specific to products and propose similarity functions tailored to them, (2) we introduce a metric for measuring the discriminative power of these similarity functions, and (3) we construct real-world test sets for two different e-commerce segments and present experimental results on them.

The remainder of this paper is organized as follows. We describe the procedure of constructing our test sets and their main properties in Section 2. Next, in Section 3, we introduce the PDP metric that we use for estimating the value of various similarity functions proposed in Section 4. We review related work in Section 5 and conclude with a summary of findings and an outline of future research directions in Section 6.

2 Data collection

Our main goals in constructing the data collection are twofold: (1) to gain insights into the differences between two e-commerce segments—electronics and toys—, and (2) to obtain a test set for evaluating product resolution techniques.

2.1 Approach

An exhaustive and statistically robust comparison would involve crawling and processing all web-shops from the segments in question. A reasonable compromise is to limit the set of web-shops considered to a certain region, for example, to a given country; this also removes the barriers of cross-language comparisons. Another rational solution to reducing efforts associated with collection building is to focus on prominent web-shops, i.e., the ones that attract many customers. Identifying all prominent web-shops of an e-commerce segment and (automatically) extracting their contents would be a stimulating research challenge on its own that goes beyond the scope of this work. As a middle ground, we chose to examine query-biased product sets: product pages returned by a web search engine in response to an adhoc product query. While this set is not comprehensive, it reasonably represents the most important web sources available for purchasing the given product. To eliminate potential discrepancies stemming from automatic product attribute extraction, we decided to extract features manually. This naturally imposed constraints on the number of products we were able to examine.

2.2 Collecting product pages

The process of constructing our two test sets is as follows. We issued a product query against a web search engine (in our experiments: Google) and collected the top N results returned (here: $N = 30$); search was restricted to a selected country (Hungary). Note that there is nothing language-dependent in our approach, the reasons for this choice were purely pragmatic (i.e., our access to data). Both test sets contain 30 queries issued by actual users. Electronics queries were selected from "popular searches" of a price comparator site. Toys queries were picked from the search logs of one of the biggest Hungarian online toy stores. Table 2 presents a few examples from the query sets (translated to English wherever needed for convenience). It is important to note that we only use these queries to collect products. The query itself is not used in the product comparison procedure.

Electronics	**Toys**
lenovo thinkpad edge 11	candamir
kingston microsd 8gb	eichhorn railway kit
samsung led tv 32"	eiffel tower puzzle 3d

Table 1: Sample queries.

Non-product pages (including results from product comparison sites) were removed manually from the result set. We also filtered out duplicates in cases when the exact same content was available under multiple URLs (as a result of various search engine optimization techniques). The remaining product pages were then manually inspected; we refer to them as the *candidate set*. Pages representing the same product were grouped together into *equivalence classes*. This manual clustering serves as our ground truth. Table 2 provides descriptive statistics.

	Electr.	**Toys**
#queries	25	30
#queries with $>$1 clusters	6	17
avg query length in characters	19.56	21.6
avg query length in terms	3.4	3.0
avg #product pages per query	4.6	5.7
min/max #product pages per query	1/8	1/11
avg #equivalence classes per query	1.24	2.66
min/max #equivalence classes per query	1/2	1/7
#different web-shops	46	38

Table 2: Statistics of the query sets.

Although we initially started with 30 queries for both segments, 5 of the electronics queries had no product pages returned within the top 30 results. Electronics queries are noticeably less ambiguous than toys queries, in terms of the number of different product clusters they exhibit. While query lengths do not differ markedly, electronics queries are inherently more specific as they always contain the manufacturer and most of the time indicate the specific model too. We also observe that in general there are less product pages returned for electronics queries. We noticed that top web search results are dominated by price comparison websites and pages presenting product reviews; these type of results were much less apparent so for toys. Future plans for improving the acquisition of product pages using web search engines are outlined in Section 6.

2.3 Extracting product attributes

We manually extracted the following attributes[4] from each product page: *name*, *price*, *manufacturer*, *productID*, *categories*, and *description*. Additionally, the followings were identified as common attributes among toy stores, thereby we collected them: *brand*, *age*, and *group* (boys, girls, or both). *Size* and *weight* information was also provided in some cases,

[4] We use product attributes and features interchangeably.

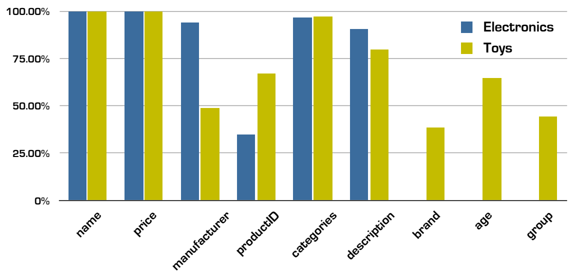


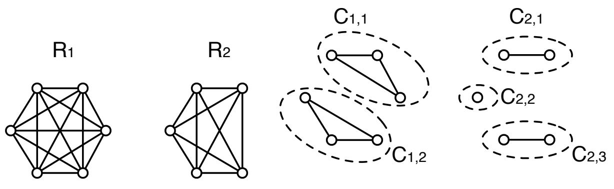
Figure 1: Statistics of product attributes. Bars show the fraction of product pages with the given attribute provided.

but for less than 15% of all product pages, therefore we did not include those features for further analysis. Figure 1 gives an overview of the presence of various attributes in product pages. Selected features are further discussed in Section 4.

3 Measuring the discriminative power of similarity functions

Our ultimate goal in this paper is to develop similarity functions specific to particular product attributes. To be able to measure how well these functions can set apart entities[5] within equivalence classes from other entities in the candidate set (i.e., the set of entities to be resolved), we introduce a metric called *pairwise discriminative power (PDP)*.

The intuition behind PDP is the following. Similarity functions cannot be compared based on the raw values they return, even if these values were normalized. Let us take an example with three entities in the candidate set, where A and B belong to the same equivalence class and C does not. If one similarity function returns $\mathrm{sim}_1(A,B) = 0.9$ and $\mathrm{sim}_1(A,C) = 0.8$, while another similarity function assigns $\mathrm{sim}_2(A,B) = 0.6$ and $\mathrm{sim}_2(A,C) = 0.3$, then sim_1 is less discriminative (hence, less useful) than sim_2 despite the higher values in absolute terms. We generalize this notion and define PDP as the ratio of the average intra-cluster similarity (i.e., similarity in equivalence classes—Figure 2(b)) over the average similarity of all pairwise comparisons in (all) the candidate sets (Figure 2(a)).



(a) Candidate sets (b) Equivalence classes

Figure 2: Pairwise similarities considered in PDP: (a) over candidate sets and (b) over equivalence classes.

Formally, let R_i denote the candidate sets that are to be resolved, where $|R_i|$ is the cardinality of the set. $C_{i,j}$ stands for equivalence classes within R_i, where $|C_{i,j}|$ indicate the number of entities in that class. $\mathrm{sim}(e_k, e_l) \in [0..1]$ is a function of similarity between entities e_k and e_l (note that it does not need to be symmetric). Note that we only consider entities to be elements of R_i and $C_{i,j}$ if sim is defined for them. The average of all pairwise similarities in all candidate sets (Figure 2(a)) is computed as follows:

$$\mathrm{avg}(\mathrm{sim}_R) = \frac{\sum_i \sum_{e_k \in R_i} \sum_{e_l \in R_i, k \neq l} \mathrm{sim}(e_k, e_l)}{\sum_i (|R_i| \cdot (|R_i| - 1))}. \qquad (1)$$

Similarly, we calculate the average of all pairwise similarities in equivalence classes (Figure 2(b)):

$$\mathrm{avg}(\mathrm{sim}_C) = \frac{\sum_{i,j} \sum_{e_k \in C_{i,j}} \sum_{e_l \in C_{i,j}, k \neq l} \mathrm{sim}(e_k, e_l)}{\sum_{i,j} (|C_{i,j}| \cdot (|C_{i,j}| - 1))}. \qquad (2)$$

Finally, PDP is defined as the ratio between the above two averages:

$$PDP = \frac{\mathrm{avg}(\mathrm{sim}_C)}{\mathrm{avg}(\mathrm{sim}_R)}. \qquad (3)$$

A PDP value close to 1 indicates that entities in equivalence classes are not that different from those in the candidate set w.r.t. the particular similarity function used. Similarity metrics that exhibit high PDP values are more discriminative, therefore, more desired.

The attentive reader might wonder why we estimate PDP using micro-averaged statistics (i.e., by considering all pairwise similarities in all candidate sets), instead of macro-averaged ones (i.e., by calculating the average similarity values over a specific candidate set and the equivalence sets formed by elements of that set, then averaging over all candidate sets). The reason is that micro-averaging is expected to result in a more robust estimate in our case, as macro-averaging would bias the results by over-emphasizing candidate sets consisting of few products or few equivalence classes (which happens to be the case for our test sets, and especially for electronics). Nevertheless, it would be worth looking at both types of averaging for a data set with more products and equivalence classes.

4 Similarity measures for product attributes

In this section we investigate means of measuring pairwise similarities between products based on particular attributes. Our main question throughout this section is this: Which product attributes possess the most discriminative power? And, do findings hold for both the electronics and toys domains, or the two display different behaviour? We introduce specific similarity functions for four main product features (name, price, manufacturer, and productID) and evaluate them using the PDP measure proposed in the previous section. In this section we will only use "local" information, i.e., data that is extracted directly from the webpages of the two products that are subjects of the pairwise comparison, and do not consider any contextual information, global statistics, or other products in the candidate set.

[5] Throughout this section we will refer to products as entities, since the metric discussed here is a generic one, applicable to any type of entity in a resolution scenario.

4.1 Product name

Name is the most obvious candidate to serve as a basis for the comparison of entities. We employ various string similarity functions that fall into two main categories: character-based and term-based. We take the raw product names, without any cleaning or other transformation step (apart from lowercasing) and use implementations of the SimMetrics Java library[6].

Character-based distance functions. We consider five edit-distance based functions. The simple *Levenstein distance* assigns a unit cost to all edit operations. The *Monge-Elkan* approach is an example of a more complex, well-tuned distance function [Monge and Elkan, 1996]. Another character-based metric, which is based on the number and order of the common characters between two strings, is the *Jaro* metric [Jaro, 1995]. We also consider an adjustment of this due to Winkler [1999] that gives more favourable ratings to strings that share a common prefix. Additionally, we use the *q-grams* approach, which is typically used in approximate string matching by comparing sliding windows of length q over the characters of the strings [Gravano *et al.*, 2001a].

Term-based distance functions. We consider five vector-based approaches, where vectors' elements are terms. *Matching coefficient* simply counts the number of terms, on which both vectors are non-zero (i.e., a vector-based count of co-referent terms). *Dice's coefficient* is defined as twice the number of common terms in the compared strings divided by the total number of terms in both strings. *Overlap coefficient* is similar to the Dice coefficient, but considers two strings a full match if one is a subset of the other. The *Jaccard similarity* is computed as the number of shared terms over the number of all unique terms in both strings. *Cosine similarity* is a very common vector based distance metric, where the Euclidean cosine rule is used to determine similarity. The cosine similarity is often paired with other term-weighting approaches, such as TF-IDF (here, we refrain from using that as TF-IDF would require global term statistics).

Results. Table 3 presents the results. The first observation is that PDP scores are much higher for toys than for electronics. Another finding is that term-based distance functions outperform character-based ones in general, although the differences are minor for electronics. These results indicate that toys belonging to the same real-world entity are likely to be named similarly, while their names differ from that of other products in the candidate set. Specifically, toys' names within equivalence classes share a lot of common terms (as witnessed by the highest overall performance of the Jaccard similarity). On the other hand, electronic products retrieved for a particular query do not markedly differ in their naming.

4.2 Price

We use the following simple function to establish similarity based on price:

$$\text{sim}_{price}(p_i, p_j) = \frac{\min(p_i.price, p_j.price)}{\max(p_i.price, p_j.price)}.$$

Distance function	Electr.	Toys
Levenshtein distance	1.0343	1.2556
Monge-Elkan distance	1.0248	1.1456
Jaro distance	1.0184	1.1508
Jaro-Winkler distance	1.0167	1.1309
q-grams distance	1.0470	1.2961
Matching coefficient	1.0511	1.3387
Dice coefficient	1.0501	1.3181
Overlap coefficient	1.0491	1.2804
Jaccard similarity	1.0602	1.4190
Cosine similarity	1.0501	1.3123

Table 3: PDP scores for product name based similarity, using various string distance metrics.

This function always returns a value in $[0..1]$ and is 1 iff the two products have the exact same price. Price-based similarity results in a PDP score of 1.0306 for electronics and of 1.1882 for toys. As with names, we conclude that toys display a higher degree of diversity in terms of price.

4.3 Manufacturer

We applied the same distance metrics as in Section 4.1 for comparing strings holding the manufacturer's name. In the interest of space we only present the main findings. Character-based and term-based distance scores are very close to each other; this is an expected behaviour given that most manufacturer names consist of a single term. The best performing functions from the two main categories are the same as for names: q-grams (PDP=1.0276 for electronics and 1.0979 for toys) and Jaccard similarity (PDP=1.0275 for electronics and 1.1056 for toys).

4.4 Product ID

Unlike with other textual fields, we do not want to allow fuzzy matches for productIDs. As IDs of products from the same manufacturer are likely to differ only in a small number of characters, applying character-based string distance functions would do more harm than good. Therefore, we use a strict string matching method for this attribute and, consequently, take similarity to be a binary function.

4.5 Discussion

Table 4 reports PDP values for the four attributes discussed (using the best performing similarity metric where more options are available), as well as the coverage in the candidate set (%R) and within equivalence classes (%C). By coverage we mean the fraction of comparisons established (provided by the availability of the given attribute for both products) out of all possible pairwise comparisons. Despite the relatively small data set, the relative ordering of attributes by PDP values are consistent across the two domains. However, PDP scores are much smaller for electronics than for toys, in absolute terms; a possible explanation stems from the fact that most electronics queries resulted in a single product cluster.

Next, we turn to individual fields, out of which product name has already been discussed in Section 4.1. As to price, we have to note that while this field is available for all products,

[6] http://staffwww.dcs.shef.ac.uk/people/S.Chapman/stringmetrics.html

	Electronics			Toys		
Attribute	%R	%C	PDP	%R	%C	PDP
Product name	100.0	100.0	1.0602	100.0	100.0	1.4190
Price	96.9	98.2	1.0306	89.6	99.1	1.1882
Manufacturer	95.3	94.6	1.0276	30.8	36.4	1.1056
ProductID	10.9	9.4	1.3333	45.0	44.2	1.9212

Table 4: Summary of the coverage and PDP values of attributes. %R and %C indicate the fraction of product pairs for which pairwise similarity could be established in candidate sets and in equivalence classes, respectively.

its content was not always successfully parsed as a numeric value—hence the imperfect coverage. Interestingly, the two segments substantially differ in the availability of manufacturer and productID attributes. While manufacturer name is available for most electronic products, this is the least distinctive feature of all. This is not surprising given that the queries used for collecting data also included the manufacturer, therefore most products in the candidate set are from the same company. Further, the identification of the specific model is often made available as part of the product name (e.g., "DeLonghi EC-8"). Conversely, toys are often labelled more creatively (e.g., "laser sword game") and keep the productID as a separate attribute. It is not unexpected that productID is the most discriminative feature of all.

5 Related work

Entity resolution (ER) is an important information integration problem that has been considered in many different disciplines (under many different names). In the Database community ER (also known as deduplication, record linkage, reference reconciliation, fuzzy grouping, or object consolidation) is the task of identifying records that represent the same real-world entity and reconciling them to obtain one record per entity. Most of the traditional, domain-independent approaches are variants of the statistical formulation introduced by [Fellegi and Sunter, 1969]; ER is viewed as a binary classification problem: given a vector of similarity scores between the attributes of two entities, classify it as "match" or "non-match." A separate match decision is made for each candidate pair. Additionally, transitive closure may be taken over the pairwise decisions. Attribute similarity measures are often based on approximate string-matching criteria [Gravano *et al.*, 2001b]. More sophisticated methods make use of domain-specific attribute similarity measures and often use adaptive approaches to learn them from the data [Bilenko and Mooney, 2003]. Utilizing the context of entities (i.e., references to other entities) brings in further performance improvements [Dong *et al.*, 2005; Bhattacharya and Getoor, 2004]. Another line of research focused on scaling ER to large databases by avoiding the quadratic number of pairwise comparisons [Baxter *et al.*, 2003; Benjelloun *et al.*, 2009]. Finally, there has been a great amount of work on non-pairwise ER, where match decisions for candidate pairs are not made independently; see [Singla and Domingos, 2006] for a generalization of these approaches using a unified framework based on Markov logic.

There are a number of problems related to *entity name resolution* within the Text Mining field. (Entity) name disambiguation (or name discrimination) is the task of grouping the representations of referents from the source documents so that each cluster contains all documents associated with each referent [Pedersen *et al.*, 2005]. Cross-document co-reference resolution is the task of determining whether an entity name (most often of type person, organization, or location) discussed in a number of documents refers to the same entity or not [Gooi and Allan, 2004]. Essentially, name disambiguation and cross document co-reference resolution are two sides of the same coin. A great deal of work has focused specifically on the resolution of person names [Wan *et al.*, 2005; Artiles *et al.*, 2005; Balog *et al.*, 2009].

Much of the research related to products focused on *mining reviews* for opinion and sentiment classification [Dave *et al.*, 2003; Cui *et al.*, 2006], summarization [Hu and Liu, 2004; Meng and Wang, 2009], and discovery and extraction of product attributes [Ghani *et al.*, 2006; Raju *et al.*, 2009]. There is not much work published on *product search*. [Nurmi *et al.*, 2008] introduce a grocery retrieval system that maps shopping lists written in natural language into actual products in a grocery store. [Pu *et al.*, 2008] develop a framework for evaluating general product search and recommender systems.

6 Conclusions and Future work

In this paper we addressed the task of product resolution in a heterogeneous environment: product pages from online stores. We conducted a data-driven exploration of using various product attributes as bases of pairwise product similarity comparisons. Further, we introduced a novel *pairwise discriminative power* (PDP) measure and performed an experimental evaluation of the attribute-specific similarity functions against the PDP metric. Our study focused on two different e-commerce segments: electronics and toys. They markedly differ in the usage of product attributes, still, our findings concerning the attribute-specific similarity functions seem to be consistent across the two.

Our work is a first step towards the ambitious task of automatic product (web)page resolution, and as such, has limitations. So far we only focused on pairwise similarity functions for a limited number of attributes (namely: name, price, manufacturer, and productID). These pairwise similarity functions are core ingredients to be used as building blocks in more sophisticated product comparison methods. There are two product-specific features that proved too complex to fit within this study: categories and description. Most webshops (as shown in Figure 1) organize their products into a multi-level categorization; these categories, however, need to be aligned. An obvious starting point is to compare term-based similarity metrics and techniques for ontology alignment [Omelayenko, 2000]. As for product descriptions, our data set revealed that these are used very differently in the two segments. For most electronics products, description entails a list of property-value pairs, while for toys, it is most often a short blurb, targeted to appeal to the customer. The two call for very different treatment.

The proposed PDP measure is a simple and intuitive one that is capable of estimating the value of similarity functions.

We recognize though that the current study lacks the validation of this measure; it remains to be tested whether higher PDP values indeed correspond to better performance when the actual clustering of products is performed.

We acknowledge that the size of our data set does not allow for a statistically robust comparison. We plan to repeat these experiments with a larger test set. We demonstrated that finding product pages using web search engines is a viable method. Initial results suggest that a more complete candidate set could be achieved by considering product comparison sites too for crawling and content extraction. We intentionally refrained from any modifications to the queries and used them unedited; as a result of that 1 out of 6 web search results were product pages, on average. This ratio could easily be improved by issuing more targeted queries. A cheap way of achieving that is to append the currency to the query, thereby excluding pages that do not contain a price. More advanced techniques might involve various reformulations of the query, e.g., by using blind relevance feedback techniques.

Acknowledgments

This work has been supported by the COMIDOR project, funded by the Norwegian Research Council.

References

[Artiles *et al.*, 2005] J. Artiles, J. Gonzalo, and F. Verdejo. A testbed for people searching strategies in the www. In *28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 569–570, 2005.

[Balog *et al.*, 2009] K. Balog, L. Azzopardi, and M. de Rijke. Resolving person names in web people search. In *Weaving Services and People on the World Wide Web*, pages 301–323. Springer, 2009.

[Baxter *et al.*, 2003] R. Baxter, P. Christen, and T. Churches. A Comparison of fast blocking methods for record linkage. In *ACM SIGKDD'03 Workshop on Data Cleaning, Record Linkage, and Object Consolidation*, 2003.

[Benjelloun *et al.*, 2009] O. Benjelloun, H. Garcia-Molina, D. Menestrina, Q. Su, S. E. Whang, and J. Widom. Swoosh: a generic approach to entity resolution. *The VLDB Journal*, 18:255–276, January 2009.

[Bhattacharya and Getoor, 2004] I. Bhattacharya and L. Getoor. Iterative record linkage for cleaning and integration. In *9th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery*, pages 11–18, 2004.

[Bilenko and Mooney, 2003] M. Bilenko and R. J. Mooney. Adaptive duplicate detection using learnable string similarity measures. In *9th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 39–48, 2003.

[Cui *et al.*, 2006] H. Cui, V. Mittal, and M. Datar. Comparative experiments on sentiment classification for online product reviews. In *21st national conference on Artificial intelligence - Volume 2*, pages 1265–1270, 2006.

[Dave *et al.*, 2003] K. Dave, S. Lawrence, and D. M. Pennock. Mining the peanut gallery: opinion extraction and semantic classification of product reviews. In *12th international conference on World Wide Web*, pages 519–528, 2003.

[Dong *et al.*, 2005] X. Dong, A. Halevy, and J. Madhavan. Reference reconciliation in complex information spaces. In *2005 ACM SIGMOD international conference on Management of data*, pages 85–96, 2005.

[Fellegi and Sunter, 1969] I. P. Fellegi and A. B. Sunter. A Theory for Record Linkage. *Journal of the American Statistical Association*, 64(328):1183–1210, 1969.

[Ghani *et al.*, 2006] R. Ghani, K. Probst, Y. Liu, M. Krema, and A. Fano. Text mining for product attribute extraction. *SIGKDD Explor. Newsl.*, 8:41–48, June 2006.

[Gooi and Allan, 2004] C. H. Gooi and J. Allan. Cross-document coreference on a large scale corpus. In *Human Language Technology/North American chapter of Association for Computational Linguistics annual meeting*, pages 9–16, 2004.

[Gravano *et al.*, 2001a] L. Gravano, P. G. Ipeirotis, H. V. Jagadish, N. Koudas, S. Muthukrishnan, L. Pietarinen, and D. Srivastava. Using q-grams in a dbms for approximate string processing. *IEEE Data Engineering Bulletin*, 24:28–34, 2001.

[Gravano *et al.*, 2001b] L. Gravano, P. G. Ipeirotis, H. V. Jagadish, N. Koudas, S. Muthukrishnan, and D. Srivastava. Approximate string joins in a database (almost) for free. In *27th International Conference on Very Large Data Bases*, pages 491–500, 2001.

[Hu and Liu, 2004] M. Hu and B. Liu. Mining and summarizing customer reviews. In *10th ACM SIGKDD intl. conf. on Knowledge discovery and data mining*, pages 168–177, 2004.

[Jaro, 1995] M. A. Jaro. Probabilistic Linkage of Large Public Health Data Files. *Statistics in Medicine*, 14:491–498, 1995.

[Meng and Wang, 2009] X. Meng and H. Wang. Mining user reviews: from specification to summarization. In *ACL-IJCNLP 2009 Conference Short Papers*, pages 177–180, 2009.

[Monge and Elkan, 1996] A. Monge and C. Elkan. The field matching problem: Algorithms and applications. In *2nd International Conference on Knowledge Discovery and Data Mining*, pages 267–270, 1996.

[Nurmi *et al.*, 2008] P. Nurmi, E. Lagerspetz, W. Buntine, P. Floréen, and J. Kukkonen. Product retrieval for grocery stores. In *31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 781–782, 2008.

[Omelayenko, 2000] B. Omelayenko. Integration of product ontologies for b2b marketplaces: a preview. *SIGecom Exch.*, 2:19–25, December 2000.

[Pedersen *et al.*, 2005] T. Pedersen, A. Purandare, and A. Kulkarni. Name discrimination by clustering similar contexts. In *Comp. Linguistics and Intelligent Text Proc.*, pages 226–237, 2005.

[Pu *et al.*, 2008] P. Pu, L. Chen, and P. Kumar. Evaluating product search and recommender systems for e-commerce environments. *Electronic Commerce Research*, 8:1–27, 2008.

[Raju *et al.*, 2009] S. Raju, P. Pingali, and V. Varma. An unsupervised approach to product attribute extraction. In *31th European Conference on IR Research on Advances in Information Retrieval*, pages 796–800, 2009.

[Singla and Domingos, 2006] P. Singla and P. Domingos. Entity resolution with markov logic. In *6th International Conference on Data Mining*, pages 572–582, 2006.

[Wan *et al.*, 2005] X. Wan, J. Gao, M. Li, and B. Ding. Person resolution in person search results: Webhawk. In *14th ACM international conference on Information and knowledge management*, pages 163–170, 2005.

[Winkler, 1999] W. E. Winkler. The state of record linkage and current research problems. Technical report, Statistical Research Division, U.S. Census Bureau, 1999.