LSH-Based Web Product Duplicate Detection Exploiting Model Words Set Adjustments

Kevin Tittel (481044)

Erasmus Universiteit Rotterdam
481044kt@student.eur.nl

Abstract. It is of considerable interest for online consumers to easily find their desired products from a vast amount of different web shops and provided products. A manner of simplifying product search is through aggregated product data across several web shops, where duplicates of products are detected reliably. Current literature suggests several methods subject to a trade-off between efficiency and efficacy. To combat scalibility issues, this paper applies Locality-Sensitive Hashing [LSH], based on sets of model words of titles with or without features including model words consisting of alphabetical characters only as well, followed by Jaccard similarity and hierarchical clustering, and examines the reliability of duplicate detection using these methods. It appears that the model without features manages to find duplicates more efficiently, and more reliably for smaller fractions of comparisons.

Keywords: Duplicate Detection · Locality-Sensitive Hashing · MinHashing · Hierarchical clustering.

1 Introduction

It is of considerable interest for online consumers to easily find their desired products from a vast amount of different web shops and provided products. A manner of simplifying product search is through aggregated product data across several web shops. Since similar products across different websites often have varying descriptions, reliable product data aggregation is a difficult task. Several methods for detecting product duplicates have already been implemented, such as Multi-Compobent Similarity Method [MSM], but are often subject to low efficiency.

Regarding the large scale of the products available on the internet, pairwise comparison for duplicate detection is not feasible. Hence, methods that reduce the number of comparisons have been introduced, such as Locality-Sensitive Hashing [LSH], based on model words stemming from the available product information like titles and key-value pairs. Next, these are paired with more advanced similarity measures such as MSM for duplicate detection. In general, the trade-off between efficiency and efficacy is of great and important interest.

This paper considers similar methods of duplicate detection where LSH is combined with the Jaccard similarity measure. A set of model words is formed

from the products' titles only, existing of either two out of three numerical, alphabetical, or special characters. Next, solely alphabetical model words are added, such as brand names and other characteristic descriptions for tv products like "led", "plasma", etc. Importantly, for each product this final set of model words is compared to both the title and the key-value pairs of features in establishing the binary matrix. Hence, this paper examines the reliability of duplicate detection based on these adjustments.

The rest of this paper is organized as follows. Section 2 is devoted to a review of related work. Section 3 introduces the methodology used in this research, followed by the results in Section 4. Finally, Section 5 concludes the paper. The code used for this paper can be found using the following link: https://github.com/KevinTittel/Computer-Science-Assignment.git.

2 Related Work

We build on previous work incorporating model words stemming from the products' provided information in order to evaluate the similarity of different products. Existing methods use model words from solely the products' titles such as the Title Model Words Method [1], or from both the titles and features, or key-value pairs, such as the Hybrid Similarity Method [2]. The Multi-component Similarity Method [MSM] [3] extends these through its hierarchical adopted single linkage clustering nature which utilizes a specific function which computes the similarity of two products. Qualitatively MSM provides significantly enhanced results, despite its scalability issues in larger data sets.

To improve upon long computation times, current literature proposes to firstly use Locality-Sensitive Hashing [LSH] to select possible duplicates before using MSM to calculate the similarity [4]. In particular, solely model words consisting of characters consisting of at least two out of alphabetical, numerical or special characters, drastically decreasing the total set of model words to be considered by the algorithm. Following this several adjustments have been made in terms of the way to select model words, such as selection model words from both titles and key-value pairs with the before mentioned alphanumerical or special characters.

This paper considers adding model words from titles only which exist of alphabetical characters only, such as brand names and other characteristic descriptions like "led", "plasma", etc. This set of model words is then compared to either the title only, or the title and key-value pair combined, of each product, in order to examine the effect of different settings on the model's performance. These model words are then used for LSH, and the standard Jaccard similarity measure for similarity detection.

3 Methodology

3.1 Data Manipulation and Representation

We clean the data to enhance the validity of the data and increase the odds of finding factual duplicates. The main reason is that different webshops deviate in

their product representations despite intending the same product. In particular, we firstly transform all upper-case characters to lower-case characters. Next, different expressions for "inch" and "hertz" are transformed into "inch" and "hz", respectively. Last, all spaces and non-alphanumerical tokens in front of the units are removed.

Given the cleaned data we establish a binary matrix of model words stemming from the products' titles. That is, we create a unique set of elements found in all products' titles, where each element has at least two of alphabetical, numerical or special characters. Next, for each TV product this set of model words of length $numModelWords$ is compared to the product's information in both title and key-value pairs. This leads to a binary matrix X of dimensions $numModelWords \times numTVs$, where $X_{ij} = 1$ if model word $i, i = 1, ..., numModelWords$ appears in either the title or key-value pairs of TV product $j, j = 1, ..., numTVs$.

3.2 MinHashing

As an input for the LSH we transform the binary matrix into a signature matrix using minHashing to circumvent the large row dimension and sparsity of the former one. In particular, we hash the binary matrix row numbers $1 : numModelWords$ to buckets of an approximately similar amount for a total number of $numHashFunc$ times. This mimics permuting the rows. For each of the $numHashFunc$ permutation we compute a signature for each of the $numTVs$ products, which boils down to the first row in order of the permutation for which the permuted X matrix finds an element 1 in the product's column. Trivially, similar products have higher odds to have the same signatures.

3.3 Locality-Sensitive Hashing

Given the signature matrix we apply LSH to obtain a set of candidate pairs for our duplicate detection. We need to tune several parameters for optimal performance. Specifically, we choose to divide the signature matrix into b bands each of r rows such that $numHashFunc = b \times r$. In each of the bands separately, the columns of TV products are hashed into buckets, where the hash function of a column is chosen by concatenating and collapsing the $numHashFunc$ signature elements in this column. In case TV products are hashed to the same bucket in at least one of the b bands, these are considered candidate pairs to be used for the duplicate detection method.

It is established that varying b and r leads to different thresholds $t \simeq (1/b)^{1/r}$ to decide whether a pair of products is considered a candidate pair. In general r dominates b, where a higher r increases t, resulting in a lower amount of false positives and less candidate pair comparisons, but a higher amount of false negatives. In case we mostly value speed and efficiency of our method, we increase r to limit the amount of candidate pairs and false positives. In case we mostly value accuracy of our method, we decrease r to screen sufficiently many candidate pairs and avoid having many false negatives.

3.4 Duplicate Detection Method

Given the output of the LSH, and the resulting candidate pairs, Jaccard similarity is used to establish the distance matrix for all products. Next, a hierarchical clustering method using complete linkage is applied to obtain clusters based on a set cluster threshold. The pairs within each cluster are supposed duplicates, which are then to be finally evaluated.

4 Results

Evaluation measures are obtained for both the performance of LSH and the duplication detection method overall. The final results are averaged over ten bootstraps, for each of which two third is used for training the algorithm and tuning the before-mentioned parameters, and one third for obtaining the evaluation measures. The data set includes for each TV product a modelID for checking whether proposed duplicates are factual duplicates, a title, and key-value pairs.

4.1 LSH performance

Evaluation measures used for the LSH are pair quality (PQ), defined as the number of found duplicates over the number of made comparisons; pair completeness (PC), defined as the number of found duplicates over the total number of duplicates; and the F_1^*-measure, defined as the harmonic mean between PC and PC ($F_1^* = (2 \times PQ \times PC)/(PQ + PC)$). Figures 1-3 illustrate the resulting PQ, and PC and F_1^* using our method as a function of the fraction of comparisons.

We compare the proposed methods with and without the inclusion of features, and including model words consisting of solely alphabetical characters as well. Regarding the PC, the results are slightly ambiguous, since for lower ratios of comparison the method without features has a higher PC than for the method with features (especially around a ratio of 0.02), whereas for higher ratios of comparison the latter outperforms the former, albeit almost equal for ratios higher than 0.3.

Regarding the PQ, the model without features performs clearly better than the model with features. The area under the curve of the model without features compared to with features increases more than double for ratios of comparison lower than 0.25, and the model without features continues to outperform the one with features throughout the range of ratio of comparisons. This can be explained by so-called overloading of the binary matrix. By using the features as a way of comparison to the set of model words, some model words may be wrongly attributed to a certain product because the item from the feature map was for a part, part of the title of that product. For example, having in the Model Word Set a '20' from 20 Watt while another TV has in the feature map a '20' from 20 inch. This will then attribute the 20 wrongly to this TV.

The F_1^*-measure again is significantly higher for the model without features, particularly for lower values of the ratio of comparisons. The F_1^*-measure achieves the highest value of 0.017 for the model without features, and 0.008 for the model with features, around a fraction of comparisons of 0.02.

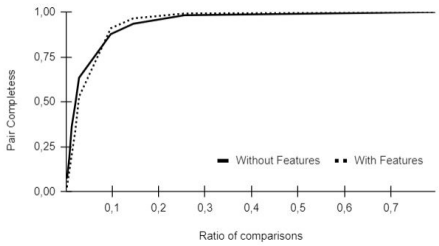


Fig. 1. Pair Completeness

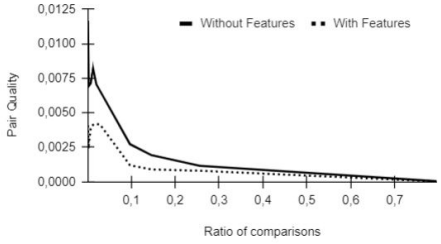


Fig. 2. Pair Quality

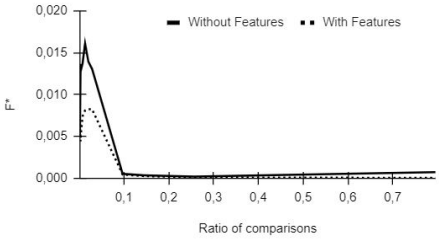
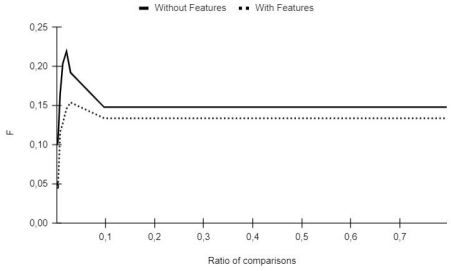


Fig. 3. F1*-measure

4.2 Duplicate Detection Method performance

Fig. 4. F1-measure against different fractions of comparisons



In order to evaluate the F1-measure as displayed in Figure 4 is used, defined as the harmonic mean between precision and recall. Similarly as before, the model without features clearly outperforms the model with features, reaching a high of 0.22 around a fraction of 0.02. Hence, the issue of scalibility is successfully combatted, since better performance is achieved for lower fractions of comparisons. Throughout the fractions of comparisons the model without fea-

tures continues to perform better. The difference between the methods is also larger compared to the F_1^* measure as given in Figure 3, making the methods more distinguishable.

5 Conclusion

In order to detect duplicates of products, this paper applies Locality-Sensitive Hashing [LSH], based on sets of model words of titles with or without features including model words consisting of alphabetical characters only as well, followed by Jaccard similarity and hierarchical clustering, and examines the reliability of duplicate detection using these methods. In particular, a set of model words is constructed from products' titles only, after which model words containing useful alphabetical only model words such as brand names and characteristic descriptions such as "led" and "plasma" are added. Next, a binary matrix was established based, where the model word set was compared to the title with or without the key-value pair features of each product. This binary matrix was transformed into a signature matrix to be used by LSH, subsequently inputted for Jaccard similarity and hierarchical clustering for detecting duplicates.

It is clear that our method without features, yet with model words containing useful alphabetical only model words such as brand names and characteristic descriptions such as "led" and "plasma", outperforms the model with features. In particular, the pair quality (PQ) and F_1^*-measures of the LSH evaluations double for the model without features compared to that with features, in particular for lower ratios of comparison, supporting the desired scalability and efficiency improvements of our model. Next, the F1-measure of the overall method's evaluation is consistently clearly better for the model without features over that with features, specifically for lower ratios of comparisons.

For future research it would be of interest to apply this better method without features, but by replacing Jaccard similarity by the before mentioned MSM method which is proven to be useful. Next, other model words of only alphabetical characters but not considered yet could be of possible interest.

References

1. Vandic, D., van Dam, J.-W., Frasincar, F. Faceted Product Search Powered by the Semantic Web. Decision Support Systems, 53(3):425–437, 2012.
2. Bakker, M., Frasincar, F., Vandic, D.: A Hybrid Model Words-Driven Approach for Web Product Duplicate Detection. Proceedings of the 25th International Conference on Advanced Information Systems Engineering (CAiSE 2003), volume 7908 of Lecture Notes in Computer Science, pages 149–161, 2013.
3. van Dam, I., van Ginkel, G., Kuipers, W., Nijenhuis, N., Vandic, D., Frasincar, F.: Duplicate detection in web shops using LSH to reduce the number of computations. In 31th ACM Symposium on of Applied Computing (SAC 2016). pp. 772–779. ACM (2016)
4. van Bezu, R., Borst, S., Rijkse, R., Verhagen, J., Frasincar, F., Vandic, D.: Multicomponent similarity method for web product duplicate detection. In 30th Symposium on Applied Computing (SAC 2015). pp. 761–768. ACM (2015)