Detecting Brands in User Search Queries

Michal Laclavík
Magnetic Media Online
360 Park Ave S, 19th Floor
New York, NY 10010
laclavik@magnetic.com

Sam Steingold
Magnetic Media Online
360 Park Ave S, 19th Floor
New York, NY 10010
sds@magnetic.com

Marek Ciglan
Magnetic Media Online
360 Park Ave S, 19th Floor
New York, NY 10010
marek@magnetic.com

ABSTRACT

In this paper, we propose a novel approach for brands detection in user search queries. The goal is to identify brands that are closely related to entities mentioned in queries. Identifying brands that the user is interested in provides valuable information for capturing the user intent and is applicable to user modeling in the domain of online advertising. We use Wikipedia as a knowledge base and a corpus for modeling user queries and for mapping mentions to brands. We first represent a query by relevant Wikipedia concepts, by an entity search approach. After retrieving relevant Wikipedia pages for a given query, we compute relatedness of these wikipages to brands from a predefined list. Our approach also generates a list of the most popular brands using Wikipedia. To the best of our knowledge, no similar approach has ever been used to model intent of queries using brands.

Keywords

brand detection, query intent, search, indexing, Wikipedia

1. INTRODUCTION

Understanding the intent of a search query is an important task in information retrieval. User search intent can be defined in numerous ways. One approach is to model user intent by Wikipedia concepts/pages [3], which we used in our work on brand detection (BD) too. When a user search query is mapped to Wikipedia pages, it can be further mapped to more specific intent signals such as category, product or brand. Capturing user intent with brands can be valuable, especially in online advertising. In the online advertising domain, the BD can help capture user interests and improve user modeling, which in turn can lead to an increase in precision of user targeting with ads relevant to their interests and needs.

Brand Detection Task: With BD, we detect queries related to a brand name or product under a specific brand. E.g., the brand `Apple` should be detected in a query such as *apple store*, where the brand name appears explicitly, but also in queries like *iPad* or *iPhone*. Similarly, if the brand `Coca-Cola` is defined, the query *Fanta* will be detected as related to the `Coca-Cola` brand. One may argue that *Fanta* also names a brand, but in our approach we focused on detecting brands from a predefined list. So, if *Fanta* is not in the predefined list of brands, and it is related to the `Coca-Cola` brand strongly, we detect it as part of the brand.

The motivation for this work came from the domain of search retargeting, a form of targeted advertising where audiences are modeled based on the search queries users conducted on websites they visited. By modeling user interests, the query retargeting has the ability to find new customers who never visited a marketer's website before. Search retargeting focuses on displaying advertisements to users who conducted searches for specific keywords or specific keyword categories in the past. For this domain, BD is an essential technique for user modeling and better user targeting.

The brand detection technique can be used in conquesting campaigns or in campaigns where marketers do not want to show ads to users familiar with certain brands. In addition, we can use BD as a feature for predictive modeling tasks (e.g. click-through rate and conversion rate prediction) as well as in reporting on user searches for the advertisers.

The task of brand detection is a challenging one. We need to map a short string (representing a query) to another short string (representing a brand) while the two have little to no lexical similarity. For this reason, research works on the subject usually extend both queries and the categories by gathering additional information, richer in textual content. The richer textual representations are then compared. A relevant method to gather additional data for queries is to retrieve results from a web search engine and then map those results to predefined brands. Alternatively, we can use existing knowledge bases where brands/products are available such as Wikipedia.

Companies focusing on the search retargeting gather large volumes of user-generated queries that need to be analyzed for identification of user search intent. The volume is often on the order of thousands of queries per second. This challenge needs a scalable and fast approach, independent of a web search engine API. We take these constraints into account in our work. Wikipedia is suitable as the intermediate corpus for query intent modeling because it is relatively small (under 40 gigabytes of text) and has very broad coverage. The proposed solution relies on two steps:

1. Extend search query with information from Wikipedia.
2. Map extended query data to pre-defined brands.

Presented at WSDM QRUMS 2016 Workshop.

We approach the first task, the query extension, as an entity search problem. We model entity representations of Wikipedia concepts. Subsequently, given a query, we retrieve entity documents best matching the user query. Similarity of retrieved entity documents are compared to list of brands using Wikipedia page similarity to return relevant brands.

The main contributions of the paper are the following:

- We propose a brand detection approach that uses limited data resources and is scalable, while reaching a high quality of achieved results, especially in terms of the precision of results.
- We propose a method which computes similarity among brands and other Wikipedia pages. The proposed similarity method reuses human knowledge encoded in Wikipedia articles. The similarity method turns query intent encoded by Wikipedia pages into brands.

The paper is structured as follows. Section 2 summarizes scholarly works related to the brand detection and extraction as well as generic approaches for user search intent. Details from query and corpus modeling are covered in [5] and in this paper we provide a short overview in Section 3. The novel brand detection approach is the main focus of the paper described in Section 4, where we provide evaluation on an decent number of search queries.

2. RELATED WORK

User search query intent is discussed in a variety of scholarly works. In this paper we mainly considered works which use Wikipedia to model search queries or model products/brands or works detecting brands in short texts.

Using Wikipedia for Query Intent Detection: In several works, queries were modeled using Wikipedia. In [3], Wikipedia data (text, link graph) are used for understanding user's query intent to model queries. However, query intent is evaluated only on three categories achieving quite high scores, acting in binary mode where the algorithm decides whether the query falls in the category or not. It is not clear how the algorithm would behave on a larger list of categories or brands. Another work [4] uses Wikipedia as a corpus for query categorization task, utilizing information retrieval approach based on vector space model, but the method achieves quite low precision scores. We have also used Wikipedia for detecting query intent turned into categories [6], where the entity search and n-gram detection approach was used for query categorization. The n-grams representing each category generated from Wikipedia have to be held in memory. This approach would result in huge memory consumption for brand detection. In BD project, we had to rethink the approach, since it would be not effective for thousands of brands; we could restrict the memory consumption for categories, with only a few hundred categories.

Brand or Product detection in queries or short strings: In [8], authors used eBay data and machine learning methods to detect brands in product titles from eBay, particularly focused on clothing brands. In our work we try to map queries to predefined list of brands, while they were able to detect new brands including typos and variations. Our methodology also attempts to detect brand name variations (and typos, if spelling correction is on), but our goal is to detect brand variations under same brand name, where the brand can act as query label or category. In [7], Wikipedia was used to detect products or brands. While this paper describes interesting experiments and lessons learned, it does not deliver real results. A similar task focused solely on brands is addressed by this paper, where we select brands list from Wikipedia by exploiting Wikipedia/DBPedia formalized data (see section 4.1).

Brand detection is a more common task in product reviews and social media [2], but especially in social media the task of identifying product or brands in short texts is very similar to the detection of brands in user search queries. In [2], the top 100 important brands were detected from microposts. The visual and multimedia were used to detect brands in addition to textual information. This seems to be the most related work to our approach.

To conclude our findings, according to the best of our knowledge, there is no related work which tries to map user search queries to a predefined list of brands, which includes also the product queries related to those brands.

3. MODELING QUERY INTENT BY WIKIPEDIA CONCEPTS

As already mentioned, in order to identify user search intent, we need to have a knowledge base for query modeling. The obvious candidate is Wikipedia, which is a publicly available knowledge base with a wide coverage. Our working hypothesis was that a large number of the queries can be answered to some extent by Wikipedia in order to get an understanding of a query, which is confirmed also by scholarly works such as [3]. There are of course brand or product-related queries not covered by Wikipedia content, but most well-known brands and products are covered.

Our solution relies on two high-level steps:

1. First, we extend the query with the information from Wikipedia corpus, by mapping the query to relevant Wikipedia concepts.
2. We map Wikipedia concepts associated with the query to items in the target brand, if any.

For the first step, the extension of query information, we took inspiration from scholarly works on entity search and we treat the challenge as an entity search problem and we successfully turned queries to Wikipedia entities by participating at *2014 Entity Recognition and Disambiguation Challenge*[1]. We have participated in the *Short Track* of the challenge, which focused on recognizing mentions of entities in a search queries, disambiguating them, and mapping them to the entities in a given knowledge base. In the ERD, our system [5] was evaluated as the 4^{th} best with an F1 score of 65.57%[2].

Recognizing Wikipage concepts in queries is the first step of our BD solution. The first step is described in detail in a related paper [5], but we summarize it briefly in Section 3.1.

The second step of the approach concerns mapping Wikipedia concepts associated with a query to target brands. Each brand is represented by a corresponding Wikipage, which is mapped manually. Having enriched query and brands with Wikipedia documents, we basically need to solve a document/entity similarity problem. We discuss the second step in detail in Section 4.

[1] http://web-ngram.research.microsoft.com/erd2014/

[2] http://tinyurl.com/ShortTrackERD14

3.1 Representing Query Intent by Wikipages

The main idea is to map user search query to related concepts (Wikipages). We will briefly describe our entity search algorithm. More details can be found in the paper [5] describing the solution we have participated with at ERD Challenge.

First we downloaded and processed Wikipedia, where we parsed more than 15 fields for each Wikipedia article including title, text, abstract, section headers, links, anchor texts of links, Wikipedia categories, or alternative names (redirects) for each Wikipedia page. We built Lucene index, where each Wikipage is represented by all those parsed fields. Some of the fields are used as search fields; others are used as alternative names of Wikipages; and some of them for the subsequent task of brand detection.

Entity search includes two main steps:

- search in the index and retrieve candidate Wikipages
- entity back-mapping - mapping entity verbalizations to the query

To briefly illustrate the process, we discuss the following example. Consider the query *Galaxy S4 vs Apple 5S smartphone* and search our index for it to get the following results (names correspond to Wikipedia page titles):

1. **iPhone 5S**
2. **Samsung Galaxy S4**
3. Smartphone patent wars
4. Samsung Galaxy S5
5. Samsung Galaxy S4 Mini
6. **Smartphone**
7. iPhone 5C
8. **Apple Inc.**
9. Samsung Galaxy

Please note that the first two search results are most relevant to query, but this is not always the case, which is why we need the second post-filtering step. The third result is also very relevant, but does not represent the entity, which is directly mentioned in the query. If we would make this third result eligible for brand detection, we would detect many more brands, as the first sentence of the article[3] mentions all important smartphone manufacturers. The search result list also contains other products which are relevant, but do not match the query intent precisely. So when we select the list of candidate entities/concepts, we then post-filter in second step. In the second step, "entity back-mapping", we are mapping surface forms of all search results back to the query. Here we use also results of Wikipedia parsing for having all relevant alternative names for Wikipages (entities) such as "S5" for "iPhone 5S". The items in bold were successfully mapped back to the query by their alternative names (i.e., surface forms). So for example, the last search result "Samsung Galaxy" can be mapped to part of the query (***Galaxy*** *S4 vs Apple 5S smartphone*), but "Samsung Galaxy S4" entity maps to longer part of the query on the same place (***Galaxy S4*** *vs Apple 5S smartphone*), so it removes the generic 'Samsung Galaxy". We are looking for longest query mapping of entity alternative names. The "Smartphone" Wikipage was also detected as an eligible result, because it maps back to the query (*Galaxy S4 vs Apple 5S* ***smartphone***).

Also, the "Apple Inc." Wikipage was correctly identified (*Galaxy S4 vs* ***Apple*** *5S smartphone*) instead of the Wikipedia page about apples/fruit, because the page contains phone-related terms, and thus the search will automatically take care of the ambiguity of search terms in most cases. To say more on disambiguation, take the example of *Apple Watch*, where we will get a product Wikipage[4]. We can detect `Apple, Inc` in the first sentence of this article, while for *apple picking* we will get "Fruit Picking" Wikipage[5] as a search result. Therefore we will clearly not detect the `Apple` brand or any other brand from the Wikipage.

When we successfully identify Wikipedia entities representing queries, they represent the user intent of the query. We can further turn them into categories [6] or detect brands representing those entities (Wikipages) as described in next section.

4. BRAND DETECTION

Our approach relies on modeling a user query with Wikipedia concepts (section 3.1) and subsequent similarity computations of query Wikipedia concepts with a Brands list. In this chapter we first define how important brands can be modeled by Wikipedia (section 4.1). Based on the preliminary brands list, we have selected more than 1,000 brands which is our human-curated list of brands we want to detect in queries. Next we provide an algorithm on how to detect brands in search queries if a query-to-Wikipedia page mapping is available.

It is also important to define what we mean by the "brand detection task". Brand should be detected in a query if the query is related to a brand from our brand list. To describe it more, we can say that a brand is relevant for a query if the brand name or product name associated to that brand is directly mentioned in the query. For example, the query *latest iPhone reviews* is related to the brand `Apple Inc.` or the query *A6 engines and performance* is related to the brand `Audi`.

4.1 Gathering Brands List from Wikipedia

In order to create the brand list, we have tried multiple approaches, such as sourcing brand lists on the web or looking at our key customer verticals. Both approaches were skewed in some way. Basically there was a need to create a representative list of brands which are relevant by brand awareness. Wikipedia could serve our needs because we could sort possible brands by its Link-In-Degree, which can be considered as a popularity raking. Our approach to the Brand selection was thus the following:

1. select all Wikipages with brand-related Infobox. The Infobox type can be gathered from DBPedia and we have thus used DBPedia data to retrieve it.
2. Sort the list by Link-in-Degree (number of links pointing to Wikipage).

The list of brand-related Infobox types was defined manually as follows: company, radio station, software, newspaper, brand, journal, magazine, airline, broadcast, publisher, hotel, beverage, restaurant, dotcom company, broadcasting network, information appliance, college, television channel, amusement park.

[3] The smartphone wars or smartphone patents licensing and litigation is an ongoing business battle by smartphone manufacturers including Sony, Google, Apple Inc., Samsung, Microsoft, Nokia, Motorola, Huawei, and HTC, among others, in patent litigation.

[4] https://en.wikipedia.org/wiki/Apple_Watch

[5] https://en.wikipedia.org/wiki/Fruit_picking

We also removed all Wikipages with less than five in degree links. With the list being quite large, we thought we could take the top 2,000 brands for use, however the problem was that some brands were represented multiple times on the list. For example `Xbox` as well as `Xbox 360` were listed; `Coca-Cola` beverage as well as `The Coca Cola Company` were listed. It would probably be possible to come up with an automatic or semi-automatic way to group these brands together, but what we ultimately decided was to manually select the top 1,000 brands, enriching the list with additional brands related to our data (such as all automotive manufacturer brands). This is how we have ended up with our list of brand names and the corresponding Wikipedia page mappings.

4.2 Computing Brands to Wikipedia Pages Similarity

The first step is to search/map relevant Wikipages to user queries. Once this is done, we need to map search results to brands. If the defined brand Wikipage, such as `Coca-Cola` is the search result itself, we just return it as a brand. The more difficult part is when the search result of the query is "MS Office" Wikipage and we want to return `Microsoft` as a brand. We need to know that "MS Office" Wikipage is related to the brand.

The first approach we have tried was to compute cosine similarity of links of each brand page and the resulting pages. This was quite good in terms of recall but poor for precision, since we were detecting all specific audio brands for generic queries such as *head phones*. Even when setting the similarity threshold high, we sometimes received those false brand results and started omitting brands which should have been detected.

We also tested an approach of finding a link in result Wikipages back to the brand Wikipage, which worked well in some cases, but would pick up also competing brands as relevant. For example, the query *rav4* would detect brands such as `Ford` and `Subaru`, since "Toyota RAV4" Wikipage links back to Wikipages representing `Ford` and `Subaru`. Also, when including link similarity together with this approach we would get the `Barnes & Noble` brand more relevant to a query for *kindle* than `Amazon`. This was because Amazon has many other product lines, and `Barnes & Noble` is more closely related to books.

After trying many approaches, we finally ended up using a very simple approach which works as follows:

1. First, we prepared all alternative names of each brand by simply taking titles and redirects of Wikipedia pages corresponding to our brands. So for example, for `Coca-Cola` we would have following list: `Coca-Cola`, `CocaCola`, `Coca Cola`. See section 4.3 for more info.
2. For each search result (Wikipage mapping to query), we take the first sentence of the article/abstract.
3. We search for alternative names from the first step in the first sentence of the Wikipage and return matching brands.

For the third step we are using an n-gram matching technique [1], which scans the first sentence and detects all the n-grams with linear complexity. We describe how we parse alternative brand names (n-grams) from Wikipedia in the next section. Since we are using n-gram matching with linear complexity (linear to length of the scanned text - the first sentence of the article), adding more brands have no effect on time-performance of the algorithm. With more brands, we simply need to keep a larger list of n-grams and larger list of brands in memory, thus the approach can be easily extended to much longer lists of brands.

Table 1: Brand detection evaluation results.

Algorithm	Precision	Recall	F1	Proficiency
Sent. & ngr.	83.04%	76.23%	79.49%	72.89%
n-grams	95.00%	31.15%	46.91%	32.00%
Sentence	81.73%	69.67%	75.22%	66.11%
Abstract	31.19%	79.51%	44.80%	66.19%
Links	6.25%	80.33%	11.60%	55.90%

4.3 Brands Alternative Names

In order to compute the similarity described in the previous article, we need to get the title and alternative names for each brand. As we already mentioned, our brand list is created as a list of Wikipedia pages related to brands. In section 3.1, we used many types of alternative names to identify eligible results by parsing Wikipedia. Here we use a smaller subset of alternative names. In order avoid having too many false positives, we only use Wikipage titles and Wikipage redirects. So for example, in the case of Apple Inc., we end up with 51 alternative names, which all link to "Apple Inc" Wikipage. To mention a few examples, we get alternative names such as: Apple computers; Apple, Inc; Apple Computer Inc; Apple.com; or even Leading the Way (corporate song).

We then can use these alternative names (n-grams) to identify brand names directly in the queries (approach described in next section) or to identify them in Wikipedia articles (third step mention in the previous section) returned as search results for the query (step described in section 3.1).

4.4 Evaluation of Brand Detection

To the best of our knowledge, there is no dataset available online which maps queries to brands. The only existing dataset is the one used in [2], however this dataset is based on social media, including image processing of brands, while our primary focus is on the detection of brands in short text (queries).

That is why we have also published the dataset with brand annotated queries as shared Google Spreadsheet[6], to facilitate reproducibility of the experiments. It consists of queries from the KDD Cup 2005 dataset that have been supplemented with hypothetical queries to represent additional products, brands, or ambiguity. The dataset is available for further research upon request over Google Docs. We have also evaluated the algorithm on an additional 10,000 manually annotated queries from our internal dataset, where we achieved similar results as reported. Experiments reported in this section are based on the shared dataset.

In the evaluation, we provide several evaluation metrics: Precision, Recall, F1 and Proficiency[7] [9].

In Table 1 as well as in Figure 1, we can see the results of evaluation, where we evaluate several approaches. The first and best approach is a combination of the second and third

[6] `http://tinyurl.com/BrandsDetectionData`

[7] `https://github.com/Magnetic/proficiency-metric`

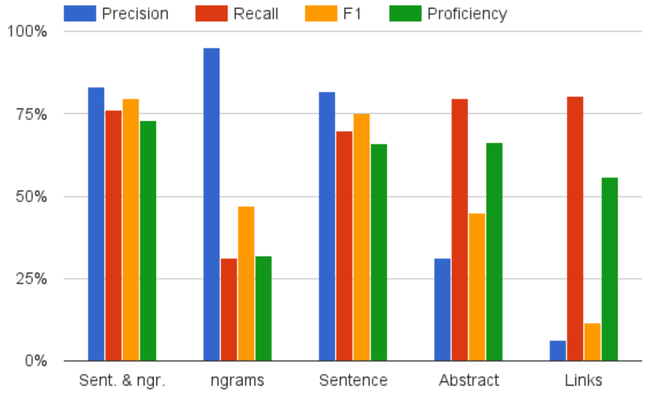

Figure 1: Various brand detection approaches

approaches (n-grams, and the algorithm described in previous section). The second 'n-grams' algorithm just searches for n-grams, extracted in the first step of algorithm in section 4.2, directly in queries without performing any search in Wikipedia Lucene index. As we can see, it has 95% precision but lower recall. The third column represents an approach which works exactly as described in section 4.2, where we search in Wikipedia for candidate Wikipages and then attempt to find n-grams in the first sentence of the article used to map Wikipages to the brand. The fourth column represents the same as third, only we have used the whole abstract to match brands. We can see that recall is higher, but precision has dropped. The last algorithm uses links of found Wikipages to map back to the brand by having a link to one of brands. Here, we can see that recall is again high but precision is extremely poor, because we return any relation to the query relevant topic.

In the Table 2, we provide an example of a few queries together with detected results and manual annotations. As you can see, it can always be discussed whether the annotation or the detected result is correct, because different humans would annotate queries differently. For example, *lineage 2* is clearly the name of a Microsoft Windows game, so it could be annotated as such. On other hand, the query *michael strahan* represents an NFL player, which detected the brand NFL and can be arguably correct. Similarly, in *camaro third generation*, the system detected both the `GM` and the `Chevrolet` brand.

Scalability: To conclude the evaluation, we should also mention the scalability of the solution for a large stream of queries. The proper evaluation of scalability is provided in paper [6] on query categorization. The brand detection is an extension of the solution for brand detection task. To summarize, when this approach was deployed in production using Apache Solr, it was able to process on average 400 queries per second on a single server. When combined with a Varnish caching server, it stabilized at around 2,500 queries because of repeating queries in real workloads. Additional boxes can be added to scale it horizontally, because the Wikipedia index can be handled with single machine.

5. CONCLUSION AND PERSPECTIVE

Understanding user query intent modeled by brands is valuable in the on-line advertising domain. In this paper, we describe a fast and scalable method for brand detection in user search queries. It is based on our previous work of entity search [5] and query categorization [6], but contains a new, unique approach for brand detection using Wikipedia, by relating Wikipedia pages to brands from defined brands list. According to the best of our knowledge, no similar solution exists. The approach can be easily extended with new brands (if available in Wikipedia) and it is scalable for thousands of queries per second. The described brand detection approach is in production at Magnetic Media Online, running as part of the Query Categorization process [6].

Table 2: Query examples with human annotation and detected results.

Query	Annotation	Detected
galaxy tab	Samsung	Samsung
michael strahan	None	NFL
camaro third generation	Chevrolet	Chevrolet, GM
girl meets world	Walt Disney	None
lineage 2	None	Microsoft

6. ACKNOWLEDGMENTS

This work is supported by Magnetic Media Online. We would like to thank Tom Comerford for editing the paper.

7. REFERENCES

[1] S. Dlugolinsky, G. Nguyen, M. Laclavik, and M. Seleng. Character gazetteer for named entity recognition with linear matching complexity. In *Proceedings of WICT*, WICT'13, pages 364–368, 2013.

[2] Y. Gao, F. Wang, H. Luan, and T.-S. Chua. Brand data gathering from live social media streams. In *Proceedings of ICMR '14*, pages 169:169–169:176, New York, NY, USA, 2014. ACM.

[3] J. Hu, G. Wang, F. Lochovsky, J.-t. Sun, and Z. Chen. Understanding user's query intent with wikipedia. WWW '09, pages 471–480, 2009.

[4] M. Kouylekov, L. Dini, A. Bosca, and M. Trevisan. Wikipedia-based unsupervised query classification. In *IIR*, pages 116–119, 2013.

[5] M. Laclavik, M. Ciglan, A. Dorman, S. Dlugolinsky, S. Steingold, and M. Šeleng. A search based approach to entity recognition: Magnetic and iisas team at erd challenge. In *Proceedings of the ERD '14*, pages 63–68, New York, NY, USA, 2014. ACM.

[6] M. Laclavik, M. Ciglan, S. Steingold, M. Seleng, A. Dorman, and S. Dlugolinsky. Search query categorization at scale. In *Proceedings of WWW '15 Companion*, pages 1281–1286. International World Wide Web Conferences Steering Committee, 2015.

[7] K. Massoudi and G. Modena. Product/brand extraction from wikipedia. *arXiv preprint arXiv:1212.3013*, 2012.

[8] D. P. Putthividhya and J. Hu. Bootstrapped named entity recognition for product attribute extraction. In *Proceedings of EMNLP '11*, pages 1557–1567, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.

[9] S. Steingold and M. Laclavik. An information theoretic metric for multi-class categorization. In preparation.