Leveraging Catalog to Resolve Conflicting Query Attributes in E-commerce Sites

Suhas Ranganath
Walmart Labs
suhas.ranganath@walmartlabs.com

ABSTRACT
Millions of people use online e-commerce platforms to search and buy products. Identifying attributes in a query is a critical component in connecting users to relevant items. However, in many cases, the queries have multiple attributes, and some of them will be in conflict with each other. For example, the query “maroon 5 dvds” has two candidate attributes, the color “maroon” or the band “maroon 5”, where only one of the attributes can be present. In this paper, we address the problem of resolving conflicting attributes in e-commerce queries. A challenge in this problem is that knowledge bases like Wikipedia used to understand web queries are not focused on the e-commerce domain. E-commerce search engines, however, have access to the catalog which contains detailed information about the items and its attributes. We propose a framework that leverages catalog information to resolve conflicting attributes in e-commerce queries. Our experiments on real-world queries on e-commerce platforms demonstrate that resolving conflicting attributes by leveraging catalog information significantly improves attribute identification, and also gives out more relevant search results.

1 INTRODUCTION
E-commerce sites are being used by millions of people to buy products in a fast and seamless manner. Users express their buying needs through search queries, and an accurate understanding of the query is necessary to return relevant items [1]. A crucial part of query understanding is to identify attributes inherent in the query [4]. For example, identifying that the query “maroon 5 dvds” has product type “dvds” helps in returning relevant items. Research on identifying query attributes in web search explores the use of semantic information [3], user engagement [5] and external knowledge bases [6]. There has been relatively less work in identifying query attributes in the e-commerce domain.

In many cases, query understanding systems have conflicting candidate attributes for a given query. In the example query, “maroon 5 dvds” the candidate attributes are the product type “dvds” the color “maroon” and the band “maroon 5”. It is not straightforward for query understanding systems to infer whether the query is referring to the band “maroon 5” or the color “maroon”. Designing algorithms to resolve conflicting query attributes can help e-commerce search systems to return more relevant items and better satisfy the buying needs of users.

This task faces several challenges. First, queries are short and contain insufficient information for systems to identify attributes. Second, knowledge bases like Wikipedia used to supplement query text in web search [2] are not focused on e-commerce domain, and can lead to insufficient and noisy information. Third, e-commerce sites have millions of users, and search algorithms have significant issues of scalability.

E-commerce search systems have access to a catalog which contains the attributes of items sold by the system. Leveraging the e-commerce catalog as a knowledge base to supplement the textual information can help to resolve conflicting query attributes. In the example query “maroon 5 dvds”, we see from the catalog that there is a significant number of items having the product type as “dvds” and have a band attribute whereas very few of them have the color attribute. This indicates that the catalog can provide valuable information which can be used to resolve conflicting query attributes. Therefore, in this paper, we propose a framework to model catalog information to better identify attributes in e-commerce queries.

Specifically, we address the following questions: How to model the catalog information to resolve conflicts in query attributes? How to evaluate the impact of the framework on e-commerce search systems? The primary contributions of the work are

• Proposing the problem of resolving conflicts in query attributes for e-commerce queries;
• Proposing a framework to model catalog information to identify query attributes in e-commerce queries; and
• Presenting evaluations of the utility of catalog information in identifying query attributes on real-world data.

The rest of the paper is organized as follows. In Section 2, we describe the proposed framework. In Section 3, we present evaluations of the framework for identifying the query attribute and its impact on ranking relevant items. We conclude in Section 4 along with possible future directions.

2 THE PROPOSED FRAMEWORK
In this section, we present our proposed framework to identify attributes in e-commerce queries. We first describe the notations used and then define the problem statement. We then use the notations to describe various aspects of the framework.

We now present the notations used in the paper. Let $q$ be the query and $\mathcal{A} = \{a_1, a_2, ..., a_n\}$ be the set of candidate attributes in the query. Let $\{\mathcal{A} - a_k\}$ be the set of attributes of size $N_a$ without the attribute $a_k$. The problem can be formally stated as follows: “Given the query $q$, attribute $a_k$, and the attribute set $\{\mathcal{A} - a_k\}$ determine whether attribute $a_k$ is present in $q$.”
We next evaluate our framework with the help of traffic weighted \( \in \{A^-n\} \) where \( n \) is the number of items in the catalog having the value \( x \) for attribute \( n \). Among such items \( m^n \) is the number of items having values for attribute \( m \). For a given attribute \( m \), we compute Eq 1 for all attributes \( n \in \{A^-m\} \), resulting in a total of \( Na - 1 \) feature values. According to Eq 1, if the query has attribute value \( n = x \), it is more likely to have attribute \( m \) if more items in the catalog with \( n = x \) also contain attribute \( m \). In the query “maroon 5 dvds”, very few items which have the product type “dvds” have values for color, and Eq 1 metric gives a lower value for \( p(\text{color}/q) \), the probability of color attribute being present in the query.

The second metric computes \( p(m/lq) \), the probability of attribute \( m \) having a value \( l \) for the query \( q \). This is formulated as

\[
p(m/lq) \propto p(m/l/n = x) \times p(n = x/q)
\]

where \( p(m/l/n = x) \) is the probability of attribute \( m \) being present where \( n = x \), and \( n_x \) is the number of items in the catalog which have value \( x \) for attribute \( n \). Among such items \( m^n \) is the number of items having values for attribute \( m \). The score is higher if more number of items having the value \( x \) for attribute \( n \) also have value \( l \) for attribute \( m \). We repeat this for all possible attributes \( n \in \{A^-m\} \) for a given attribute \( n \) resulting in an additional \( Na - 1 \) feature values. The value set for a given attribute follow a power law distribution where few values are prominent, so we employ log smoothing to make the values linearly distributed.

We integrate the scores derived from the catalog metrics into a feature set. Our metrics are scalable and hence suited for handling large-scale traffic common in an e-commerce site. We use an out of the box classifier on the feature set to determine whether or not the given query has the attribute value.

<table>
<thead>
<tr>
<th>Dict Lookup</th>
<th>Framework</th>
<th>Gain</th>
</tr>
</thead>
<tbody>
<tr>
<td>+6.48%</td>
<td>+11.37%</td>
<td>+8.3% +5.36%</td>
</tr>
</tbody>
</table>

Table 1: Performance of the framework on attribute identification and ranking

**REFERENCES**


