What Makes a Phone a Business Phone

Querying Concepts in Product Data

Silviu Homoceanu, Wolf-Tilo Balke
Institute of Information Systems
Technische Universität Braunschweig
Braunschweig, Germany
{silviu, balke}@ifis.cs.tu-bs.de

***Abstract*— The Web has become the primary source of information containing both structured and unstructured information. A good example is e-commerce where products are usually described by technical specifications (structured data) and textual user reviews (unstructured data). Both sources of information complement each other, covering quantifiable as well as perceived aspects of each product. In fact, for most searches users will have more or less abstract concepts in mind, as opposed to clear cut categorical information. In this paper we develop a novel approach to reveal implicit product features for querying by combining structured product data with natural-language product reviews. Using a self-supervised learning technique we progressively build a query-aware representation of the product domain under consideration. This representation can then effectively be used for intuitive querying. We performed extensive experiments confirming the effectiveness of our approach over real world product data. In particular, our evaluations show vastly improved precision and recall over the respective IR techniques.**

Keywords: product search; e-commerce; query expansion; semantic mismatch; concepts; concept extraction.

I. Introduction

During the last decade the Web has evolved into the prime information source for a vast variety of topics, particularly fostered by the large amount of user generated content. In the wake of information searches also services like e-commerce have rapidly expanded: read about something – buy it! Indeed more and more products or services are marketed and sold through online platforms today, and offering a high quality portal is a distinctive advantage in competition. To improve customer experience, popular online shops like e.g., Amazon.com or DooYoo, provide users with extensive information regarding products, like technical specifications, expert reviews, and relevant user comments or ratings. The common point of reference is the entity (e.g., product, music or video file, person,…) in question, but due to the mixed nature of the content almost all platforms have to cater for different types of data namely structured data (like product databases) and unstructured data (like product reviews).

However, when it comes to querying such mixed data today's platforms still face a difficult problem. Most platforms offer navigational interfaces for SQL-style access to structured data (via categories) and then a simple keyword style search for the actual values. Some even offer IR-style keyword search on user comments or product description, but a viable integration of search is still lacking. As an example consider querying Amazon.com for a 'Nokia E72' cell phone. Easily enough the result is the entity matching the search criteria as in classical databases. But real world user queries tend to be more complex: often certain aspects of an entity's purpose or application are the focus of queries like a 'business cell phone', a 'city car', or 'music for a wedding'. But trying for example the 'business cell phone' query in Amazon.com, the system only returns 5 devices, entirely missing out on business phone market leaders[1] like the BlackBerry Bold 9700, the Motorola Droid, or the HTC Touch Pro2.

How can this happen? The major reason for the catastrophic recall is that whenever a query term is not explicitly mentioned in the stored data, today's systems are not able to interpret the intended information need. As [1] points out, there actually is a *semantic mismatch* or *gap* between the data presentation and the user perception over entities. In a nutshell, crucial implicit information like 'what explicit features make a cell phone a good business phone?' cannot be derived and this semantic mismatch is unfortunately present in most entity-centric searches.

As a running example for the rest of this paper we will stay with the domain of cell phones. First, this segment of consumer electronics is well-understood, and what is more, it offers plenty of real-world data since according to Gartner mobile phones sales (especially smart phones) currently is one of the strongly growing markets[2]. In any case, queries on implicit information are not only typical in the cell phone domain, where discussion boards regularly refer to flowery categories like 'ideal for social networkers', 'perfect for

---

[1] Cha, B., (Jan. 2010). Best smartphones for business users. CNET, http://reviews.cnet.com/4321-6452_7-6544038.html

[2] http://www.gartner.com/it/page.jsp?id=1372013

fashionistas', 'tough as nails', or 'multimedia marvels'. But the basic problem is also consistent with recent results from other domains like the predominant tagging of explicit media features, in contrast to the high number of queries on implicit (usage-based) features in online image repositories or music stores (see [2]). Therefore, being able to transform implicit information needs into explicit terms for querying is generally of vital importance for building successful e-commerce platforms.

The challenge of implicit information needs vs. explicit queries has been discussed before and is directly addressed by some retrieval paradigms. However, experiments on real world data have shown that classical IR techniques like the vector space retrieval model (VSM [3]) and latent semantics (LSI [4]) do not achieve satisfying results [5]. On the other hand query expansions, i.e. augmenting user queries with relevant semantically related terms, show promising results, if only the expansion terms are chosen in a sophisticated manner. While first approaches only focused on synonymy and term disambiguation, today domain knowledge is incorporated. Expansion algorithms range from using simple lexical databases [6] like WordNet, existing domain ontologies [7] like MeSH (Medical Subject Headings - a controlled vocabulary thesaurus used for indexing medicine related articles) to extracting language models from the text like probabilistic models based on term co-occurrence [8]. Following our running example, a clear semantic connection between the 'business cell phone' concept and technical features like 'email clients', 'organizer', 'calendar', 'notepad' and 'file browser' could be established.

In this paper we present a novel query expansion method, which is able to solve the expansion problem for entity centric searches by bridging structured and unstructured data, with the help of a self-supervised learning technique.

For evaluation purposes we used cell phones with real world user reviews crawled from the Web, which were subsequently tagged by domain experts with respect to prevalent concepts in the domain. We evaluated our algorithm in terms of precision and recall against two standard IR methods, LSI and VSM, as a baseline. Our experiments show impressive improvements on average maintaining a precision over 0.8 for about 25% recall, and precision values greater 0.5 even for recall values of 90% in contrast to less than 0.2 for comparable VSM and LSI approaches.

II. DEFINING THE RETRIEVAL TASK

In this section we will lay the foundation of our approach by briefly presenting the application query type and formalizing the actual problem.

A. Revisiting Entity-Centered User Queries

As argued before, every query expansion technique reaches its greatest benefit when query terms refer to concepts, which cannot be queried with simple SQL or IR techniques. But is there really a need in today's information portals for specifically answering concept-specific queries? Studying the AOL Web search query logs (comprising logs of all searches done by 650,000 AOL users over the course of three months in 2006), with regard to our sample domain of cell phones, we observed that the amount of queries on *clear* product features focusing on capabilities e.g., 'camera' or 'voice dial', is actually smaller than the amount of queries on concepts mainly focusing on usage e.g., 'cell phone for kids', 'cell phone for seniors', or 'business cell phone'.

In particular, we extracted 21,650 cell phone relevant entries through the use of regular expressions. After manual inspection we classified all queries into six base categories (see Fig. 1) using a string similarity measure. The resulting categories deal with:

- **Products:** represents about 22% of the queries. It contains queries related to brands, product prices, product features, specifications, and types. E.g., 'Motorola Razr', 'cell phone battery', or 'cell phone for kids'.
- **Telecom & Pricing Plans:** for example 'Verizon cell phones' or 'compare cell phone plans'. This category represents about 30% of the cell phone related queries.
- **Accessories:** Represents 17% of the queries, and refers to products for cell phones e.g., 'sexy phone wallpaper' or 'ringtones'.
- **Phonebook:** 13% in size refers to cell phone numbers, or reverse phonebook lookups, e.g., 'cell phone number lookup'.
- **Unspecific Queries:** about 15% of all queries representing too general queries, usually simply 'cell phones'.
- **Other:** about 3% containing more exotic queries like 'help finding lost cell phone', or 'cell phone health risk'.

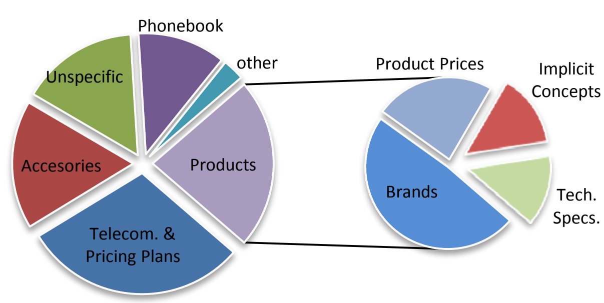

Figure 1. AOL Query Log.

Focusing on the category with references to products, we observed that the majority of queries are either concerned with a specific brand (e.g., 'Motorola' or 'Sony Ericson phone') or the price (e.g., 'Nokia 5300 price' or 'cheap mobile phone'). Still, about 27% focus on specific product features and the amount of queries for concepts (like 'multimedia phone' of 'cell phone for seniors') surprisingly, is about 14% of the total feature related queries. Although they represent only a relatively small percentage, answering such queries is vital for the search process. The reason lies mainly on the consumer buying process. According to Engel et. al. [9], [10] users generally first gather information in a task-based manner, i.e. they try to identify best products for the intended usage. The second phase is purely informational. Here users compare technical specifications of candidate products, or prices from different vendors. Since informational queries usually are posed several times for different products or for different vendors, it accounts for the significant difference in percentages.

Selecting the appropriate technique to correctly answer a query needs a more detailed distinction of the query terms. As we have shown in [11] the main differentiating factors are the *clarity* of the query term and the level of *user consensus* over the expected result. These factors span a design space, where typical query terms can be arranged. Considering the techniques for evaluating the respective queries we find that *clear* queries can be answered successfully with simple SQL-based techniques directly relying on product databases. In contrast, for *unclear* query terms techniques from the field of IR evaluating text e.g., product descriptions or reviews would be more appropriate. However, focusing our attention more on the queries in-between, currently there are very few suitable approaches for query evaluation, because a combination of structured/unstructured information has to be exploited for retrieval. Supporting those queries in a satisfying manner is the subject of our research.

In this paper we focus on answering conceptual queries with large *user consensus* over the expected result, leaving preference based conceptual queries like ‘best’, ‘beautiful’, etc. as a subject for future work.

B. Bridging the Semantic Gap

Although many real world queries use conceptual information, it is difficult to define what a concept actually is, and how it can be reliably spotted in queries. Psychology [12] defines concepts as a cognitive unit of meaning, typically associated with a single meaning of a term. Any term can therefore be the representation of a concept. The major importance of specific concepts in practical life comes with the generally consensual notions humans connect with some concepts: each concept carries connotations that immediately create an intuition about what is meant, and thus enable efficient communication. For example asking about a ‘cell phone for kids’ will immediately bring up ideas like robustness, ease of use, fun colors, security features, and parental control pricing plans. Explicitly adding exactly these connotations to a query is what makes applying a semantic query expansion technique so promising for good retrieval quality.

However, lacking a clear definition, detecting conceptual features in queries is a serious problem. Whereas for *explicit features* like ‘weight’, ‘size‘, or ‘display type’, new developments in declarative query languages already allow a mapping of previously unknown attributes to actually existing attributes in the underlying data (e.g., using malleable schemas [13]), the recognition of *implicit conceptual features* like ‘portability’ or ‘design’ is much harder. Still, even if implicit conceptual features cannot be clearly defined and the exact disambiguation is beyond the scope of this paper, preparing the underlying data collection to answer at least the most often occurring implicit queries is of strategic advantage.

Therefore, in this paper we will rely on a few simple, yet suitable, heuristics. First, implicit features obviously can never be attribute names of structured data, and also in the respective set of values they should rarely occur. Similarly, in unstructured text documents an implicit concept should occur not too often, either. But since texts are the usual way to communicate connotations and tie concepts to entities, any important implicit concept definitely should occur at least sometimes. In the following we consider any noun (<*N*>) and nominal phrase (<*NP*>) in a query as implicit conceptual feature that complies with *Observation 1*.

Observation 1: *Implicit Conceptual Feature*

Let x be any query term, $f_S(x)$ be the percentage of entities for which x occurs in values of structured data, and $f_D(x)$ the percentage of documents grouped by entities explicitly mentioning x.

An *implicit conceptual feature* q is any query term for which $0 \leq f_S(q) \leq r$ and $s \leq f_D(q) \leq t$, where r, s and t are domain specific parameters.

For our cell phone example and the later experiments, we tried different values for r, s, and t. We found that occurrences under 5% in structured data and occurrences in between 2% and 10% of unstructured reviews are sufficient for detecting most implicit conceptual features without generating too many false positives. As stated above these parameters are collection-specific, and will have to be adjusted for other data sets.

Now we are ready to address the problem of answering implicit conceptual queries. As stated above, nowadays there are large amounts of entity-related data available online, but usually comprising both structured and unstructured data. But actually for our retrieval task this is not a problem, but rather a feature. This is on one hand because most concepts will only be explicitly tied to entities in unstructured texts, thus descriptive vocabularies can be derived by co-occurrence analysis. But many concepts are also to some degree affected by certain structural characteristics, thus the statistical analysis and exploitation of value distributions can point to similar entities (e.g., ‘portable’ items will definitely show a bias towards smaller sizes and lighter weight). In fact, starting with a seed vocabulary for some relevant entities, and learning their structured characteristics to find similar entities, which in turn are used to expand the vocabulary and learn even more about the structural bias, will lead to a cyclic improvement of a model that subsequently can be used for effective querying.

In summary, for implementing the query expansion of some initial implicit query term our approach needs the extraction of terms relevant to some concept from the underlying data, and thus has to bridge the gap between structured and unstructured information. The retrieval task can be formalized as follows:

Problem Statement: *Query Expansion for Implicit Features*

Given: A relational database S containing data with respect to entities $P_1, \ldots, P_N$. For each entity P_i, there also are text documents $D_{i,1}, \ldots, D_{i,n(i)}$ describing P_i.

Task: Given a user query Q containing an implicit conceptual feature q, derive an expanded query $Q' := q \cup \{q_1, q_2, \ldots, q_k\}$, where $q_1, q_2, \ldots, q_k$ are terms from S and D which explicitly describe q (with corresponding weights $w_1, \ldots, w_k$).

III. THE QUERY EXPANSION PROCESS

The problem of querying for implicit conceptual features is typically solved by using a query expansion technique. The key task however, is the selection of the right terms for expanding the query. An intuitive approach would be to consider for the expansion all the terms occurring together with the queried concept in the *product data*. (Note that by product data we understand structured and unstructured data i.e. technical specifications and respectively product reviews). But the number of such terms is quite high, and although the query expanded in this manner leads to high recall, the precision is catastrophic with almost any product qualifying as a result. In consequence, we first choose a set of candidates that appear together with the query in product data, then we calculate the weight of each candidate term based on a function similar to the term co-occurrence and finally we select only those terms with the highest weights.

A. Choosing the Candidate Terms for Query Expansion

The query expansion is performed based on a corpus of products. Each product in the corpus is described through one or more text documents and one tuple in the technical specifications table. Of course any term appearing in documents or the tuples could be of interest for the expansion. But particularly in the case of the documents, it's obvious that many of the terms have no relation with the queried concept. Actually when referring products, concepts mostly relate to some product features. *Thus any term expressing a product feature and co-occurring with the queried concept in the product data is considered a candidate for the query expansion.* Two steps have to be performed for choosing the candidate terms: first, select the query relevant product data (data in which the queried concept appears) and second extract the product features from the selected data.

1) Query relevant product data

A document is likely to be relevant if the query is mentioned in it. But not the same can be said about the structured data. We have found numerous cases where a product manufacturer would include some task based concept in the product name, model or series, although the product is not a good candidate for the concept. However, if the query is explicitly mentioned in a document, then the technical specification of the corresponding product is also relevant with respect to the query. In consequence, the product data can be separated into classes c (highly relevant data with respect to the query) and $\bar{c}$ (the remainder of the data) starting from the documents. c then comprises documents (D_c) and tuples from the structured data (S_c) relevant with respect to query q:

$$D_c = \{d_i \mid d_i \in D \wedge \exists d_j \in D'_c \text{ s.t. } cos(d_i, d_j) \geq \theta\},$$

where θ is a collection specific parameter regulating the precision of c, and *cos* represents the well-known cosine similarity measure between two documents represented as vectors in the VSM;

$$D'_c = \{d_j \mid d_j \in D \wedge d_j \text{ contains } q\};$$

$$P_c = \{p_i \mid p_i \in P \wedge \exists\, d \in D_c\ with\ d\ describing\ p_i\};$$

$$S_c = \{s_i \mid s_i \in S \wedge \exists\, p_i \in P_c \text{ s.t. } s_i\ tech.specs.of\ p_i\}.$$

For separating the documents we have used VSM with TF-IDF [24]. As a similarity measure between documents and the query we have used the cosine metric. Accordingly $\bar{c}$ comprises $D_{\bar{c}} = D - D_c$ and $S_{\bar{c}} = S - S_c$.

2) Product features

In the case of unstructured data product features are usually represented through nouns and nominal phrases [19]. Some adjectives can also imply product features e.g., 'heavy' may imply the 'weight', but these are rather infrequent cases. Consequently, in order to extract the candidate terms, we have applied standard natural language processing (NLP) techniques like part-of-speech tagging (POS) and chunking. Word inflections have been eliminated by means of stemming.

In structured data, products are described through table attributes and the corresponding values. While all attributes are product features, from the values we have only considered the ones corresponding to categorical attributes. Obviously all values in the product table define a certain aspect of the product but the categorical attributes bear most of the differentiating force. Typical examples of such values are 'nokia', 'apple', etc., for the 'brand' attribute, or 'candy bar', 'clam shell' for the 'form factor' attribute. Numerical values like in the case of the 'price' or 'weight', have dynamically been reduced to the ordinal values 'low', 'average' and 'high'. We have established the 'average' interval of the values for an attribute as being between [average of the values – one standard deviation, average of the values + one standard deviation]. We have then set the 'low' and 'high' intervals accordingly. Although they are not candidate terms, together with binary values, the ordinal values play a significant role in calculating the weight of their corresponding attributes, as discussed in the next subsection.

Finally, after establishing what *product features* and *query relevant product data* stand for, we can formally define the set of candidate terms:

Definition 1: *Candidate Terms (CT)*

Let CT_D and CT_S be the set of query expansion candidate terms from documents and respectively structured data, with:

$$CT_D = \{t_i \mid (POS(t_i) = <N> \vee POS(t_i) = <NP>) \wedge (t_i \subseteq d, \text{ with } d \in D_c)\} \text{ and}$$

$$CT_S = \{t_i \mid (t_i\ table\ attr.from\ S) \vee [(\exists a_{t_i}\ attr.of\ t_i) \wedge (a_{t_i} \in \{categorical\}) \wedge (t_i \subset s, \text{with } s \in S_c)]\}$$

where, *POS* (t_i) represents the part of speech of term t_i, and the <*N*> and <*NP*> tags represent the noun and respectively nominal phrase parts of speech.

We define the set of *candidate terms* as: $CT = CT_D \cup CT_S$.

B. Calculating the Weight of Candidate Terms

Associating the candidate terms with the *right* weights is crucial for the entire process. The weight of a term must reflect the term's contribution to describing the queried concept.

In this paper we estimate the weight of a candidate term by using an approach from the field of *document classification without negative examples revisit* [22]. The basic idea is to give a higher weight to candidate terms which appear quite often in data from c and not that often in data from $\bar{c}$.

Definition 2: *Weighting Function (W)*

Let ct_i be any candidate term from *CT*. $n_c(ct_i)$ and $n_{\bar{c}}(ct_i)$ represent the number of documents or tuples that contain ct_i from c and respectively $\bar{c}$.

The weight of ct_i, denoted $W(ct_i)$ is estimated by calculating the difference between the normalized frequency of ct_i in c and $\bar{c}$:

$$W(ct_i) = \frac{n_c(ct_i) - min_c}{max_c - min_c} - \frac{n_{\bar{c}}(ct_i) - min_{\bar{c}}}{max_{\bar{c}} - min_{\bar{c}}},$$

where max_c and min_c are the maximum and minimum value of $n_c(ct_i)$ for $ct_i \in c. max_{\bar{c}}$ and $min_{\bar{c}}$ are the maximum and minimum value of $n_{\bar{c}}(t_i)$ for t_i being a product feature from c or $\bar{c}$.

NB: for the candidate terms representing table attributes, the weight of an attribute is calculated based on the corresponding value, with ct_i being extended in this case to the attribute-value pair. For attributes with numerical values, they must previously be transformed to ordinals as presented in subsection *A.2*. To clarify, the weight of the 'price' attribute for example will be calculated as the maximum out of three weights, one for 'low price' one for 'average price' and one for 'high price'.

The key factor is that the weight of each term is normalized with respect to typical terms from both c and $\bar{c}$. This is critical because $|c| \ll |\bar{c}|$. In this way important candidate terms with implicit connection to the queried concept aren't severely penalized despite appearing also in $\bar{c}$.

But since we have split the product data into two classes why not train a classifier to deal with new products? As argued in [16] and as shown in the evaluation section, classical IR techniques like VSM are not able to retrieve many of the eligible products. Therefore it is most likely that both $D_{\bar{c}}$ and $S_{\bar{c}}$ contain data which is implicitly relevant regarding the query. For this reason, classifiers like SVM or decision trees are not the right option (see [22] for further details). Furthermore, typical weight measures associated with *discriminative feature weighting* like term co-occurrence, mutual information or information gain tend to excessively penalize important terms due to the noisy classification.

C. Selecting the Expansion Terms

Having calculated the weight of all candidate terms we are now ready to choose the most appropriate terms for query expansion. Taking a closer look at the weighting function, the candidate terms are associated values between [-1; 1]. As intuitively expected, there are few very week candidate terms, with weights close to -1, many general terms, with similar normalized appearances in c and $\bar{c}$ and weights close to 0, and some strong candidate terms with values closer to 1. For the query expansion, we have chosen the candidates with the highest weights according to the 'three-sigma rule' (average plus three standard deviations).

IV. Evaluation

In this section we present the methodology we have used for evaluating our approach. We first introduce the metrics and the baseline method used as reference, in Section A. In Section B we describe the data used for the tests, while the discussion of the obtained results is performed in Section C.

A. Evaluation Methodology

Query expansion is a classical method for improving the retrieval performance of IR techniques. For evaluating purposes, it is only natural to compare the results it achieves against the well-known VSM featuring TF-IDF with cosine similarity. Since our queries contain implicit concepts and LSI is a promising technique which indexes and retrieves documents in a low-dimensional concept space by making use of semantics, we have considered it also as an important reference for our tests. As expected, the metric for the evaluation is represented by the Precision/Recall [25] curves.

As test set we have used documents describing products, which were subsequently tagged by domain experts with respect to prevalent concepts in the domain. For each conceptual query, expansion terms have been extracted based on a product corpus comprising both structured and unstructured data, as presented in Section III. The query expansion terms have been used to compute the relevance of each document from the test set. *The relevance of a document regarding an expanded query is given by the sum of weights of the query expansion terms, the document contains.*

B. Data

For our tests we have analyzed data sets from the field of cell phone domain. If when it comes to the structured data, the only possibility is to use technical specifications of the products, text documents come in more flavors like for example editor's reviews, user reviews or blogs. Analyzing these information sources we have observed that they offer different perspectives of the products. If the editor's reviews present the features and facts in a more objective manner, with extensive but field-relevant vocabulary, the user comments are smaller in size, concentrate on a reduced number of features, and are strongly influenced by the user's interests and point of view towards the entity. Blogs tend to be even more emotional than user reviews making sentiment analysis an absolute requirement. Sentiment analysis however remains very unreliable when the text uses slang, sarcasm, emoticons, prolonged letter usage, capitalization, punctuation, etc. For this reason, we have performed the query expansion process on a collection of 350 products with the corresponding technical specifications and 500 editor's reviews. The data has been crawled from phonearena.com, a top Web publication in the field.

The quality of the expansion terms has been tested on the more challenging user reviews, to question suitability of the expansion terms for non-expert typical user language. The test set collection, comprises 200 user reviews regarding the latest cell phones, we have crawled from CNET (http://www.cnet.com - a leading technology oriented Web site offering large amounts of both editor and user reviews for different products). These reviews have then been manually labeled by experts in the field, as either being relevant or not with respect to three most important[3] test features: 'business', 'social networking' and 'camera', for cell phones. We have chosen these three features to show the differences which occur when considering rather ambiguous, classical concepts, here the 'business' concept, emerging concepts like 'social networking' and finally merely technical characteristics such as the 'camera' feature.

C. Discussion of the Results

First we have tested both VSM and LSI with the available data. To our surprise, LSI always obtained poor results even when compared with the VSM (see Figure 2.). Different numbers of dimensions have been tried for the LSI (10, 20 and 100 dimensions –see [4]) for all our test scenarios, but the results still showed a poor quality. The reason for this behavior is the small amount of data which is available for training the LSI. The collection of 500 documents seems rather limited for the latent semantics needs. Editor's reviews are rather scarce resource, so we then increased the document base for LSI to 6000 documents, supplementing with user reviews. However, as previously stated, user generated documents do not offer similar advantages as editor's reviews do. Even with this large collection, LSI is still not able to achieve notable results. Collecting editor's reviews over long periods of time is also not a solution and cell phone domain is a great example in showing how fast concepts evolve with time. Taking this into consideration, in the following we only show graphs of our approach against the TF-IDF based VSM.

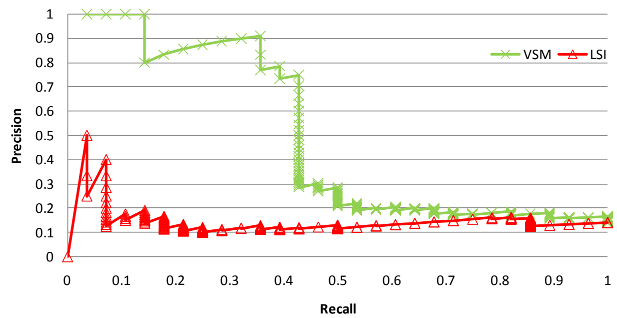

Figure 2. 'Business' - LSI vs. VSM.

1) *The 'Business' Concept*

The query expansion comprises terms which have orthogonally been extracted from structured and unstructured data. But is there a real need to use both of the underlying sources? To answer this question, we have evaluated the results we have obtained by first expanding the query with just terms originating from each of the data sources and then combined. In Tab. TABLE I. we present the query expansion terms extracted from the technical specifications, along with the top 10 out of a total of 153 terms extracted from the unstructured data.

TABLE I. QUERY EXPANSION TERMS

Structured data	Unstructured data
phone Type	windows mobile
smart phone	business
phonebook features	work
picture id, multiple numbers	letters
Phonebook	notes
phonebook capacity	fields
	qwerty keyboard
	navigation
	outlook
	task

As shown in Figure 3. , expanding the query only on the technical specifications (SQE), leads to poor results in terms of precision and recall. On the other hand, VSM achieves good precision up to a recall of about 40%. Above this threshold, the precision heavily deteriorates. The behavior of VSM becomes clear after analyzing the data: 43% of the reviews the experts have labeled as relevant towards the 'business' concept, explicitly mention the conceptual feature, leading VSM to identify with a high precision exactly these documents. Also worth mentioning is the 'saw-tooth shape' effect [26], common for VSM in precision/recall curves. The same test performed with the expansion terms from the unstructured data (UQE) already delivers results superior to VSM. While the results show a slightly worse precision for

[3] http://tech.uk.msn.com/features/photos.aspx?cp-documentid=149711759

low recall values, it quickly outperforms VSM for the documents where the concept is implied.

Finally, since the structured and unstructured data cover different aspects of products, results are further improved by exploiting both of sources (QE line in Figure 3.). Not only does the precision for low recall values drastically improve, but it is also maintained above 50% up to a recall above 90%. Also worth mentioning is the fact that at 100% recall we still have a precision of approximately 40% compared to values under 20% for the VSM approach. This is also consistent with the behavior we have observed in the case of the other concepts. Following this argument, we present only the combined model in the remaining evaluation tests.

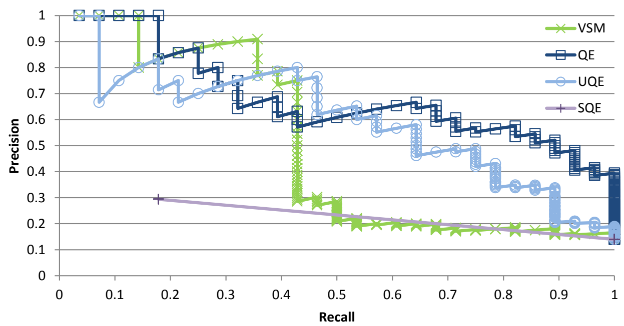

Figure 3. ‘Business’ – Query Expansion vs. VSM

2) The ‘Social Networking’ Concept

The ‘social networking’ concept is an exceptional example of how the syntactical representation of concepts can be misleading. From a linguistic perspective this concept is represented by a nominal phrase with strong syntactic relation to the ‘networking/network’ technical feature. This relation however doesn’t reflect human perception. For instance, the concept of ‘social networking’ and the ‘UMTS network’ technical specification show no semantic connection whatsoever. In such cases, the concept is validated by *Observation 1*since ‘social networking’ doesn’t appear in technical specifications, and it is only mentioned few times in the reviews. But the VSM based technique used for establishing D_c (Section III.A.1) will also retrieve a high number of documents regarding the ‘networking’ feature. And especially in this case since a technical feature is characteristic for all the products, this shifts the focus of the query expansion process towards an undesired direction. In the case of concepts comprising multiple words, we prune the documents that contain only some of the words forming the concept. D_c then comprises documents either containing the complete concept or not at all. This ensures that the concept itself is expanded and not some of the words forming it.

As shown in Figure 4. , our results are again, even in this tricky case indeed much better than the ones achieved by the VSM. The curve is also different from the ‘business’ concept. This is due to the fact that more user reviews share the same strength, i.e. the recall is improved without significantly lowering precision. Actually, it is a consequence of the reduced number of terms selected for query expansion, which characterize this concept.

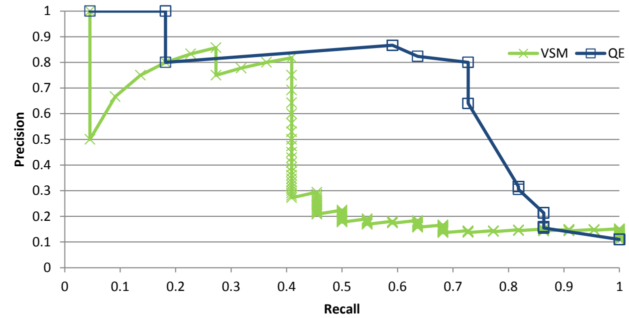

Figure 4. ‘Social Networking’ as Single Concept

3) The ‘Camera’ Technical Feature

Finally, inspired by the contradicting terms obtained when considering also the ‘network’ feature as seed for expansion, the last of our tests, investigates a query purely based on a technical feature. The results show that our approach is at the present time indeed limited to expanding implicit conceptual features (see Figure 5.). The retrieval performance for technical features is merely comparable to the standard vector space model. The reason is that technical features are always explicitly mentioned in most of the editor reviews, as well as the technical specifications, regardless of the product. For example, the ‘camera’ technical feature is present in 80% of the documents from the collection used for query expansion. This clearly calls for standard techniques and our approach cannot offer any additional benefits here.

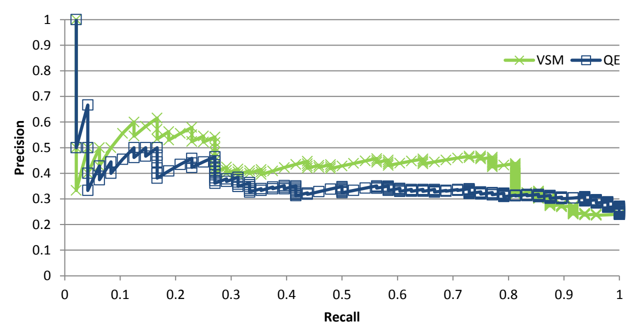

Figure 5. ‘Camera’ Feature.

V. Related Work

Recently, several search engines have been proposed, which can retrieve products even if the query keywords do not match the product tuples in the database [14][15]. Such engines extract the entities which co-occur with keywords from the query, in documents on the web. But for concept driven querying this approach is likely to suffer from *incompleteness* since most of the concepts are mentioned only in a few documents. The reason is that concepts are rather implied by means of related terms. We tackle this problem by further expanding the query with terms related to the concept. Such search engines may also suffer from *impreciseness* of the results. In some of the documents the concept may be present but with a different meaning then the one intended by the user. Searching for a 'business' cell phone, one would also encounter cell phones with a description similar to '…it has GPS, so you can locate businesses nearby!'. By adding weights to the query expansion terms we are able to maintain a greater precision even for high recall.

On the other hand, approaches like [16][17] follow a query transformation technique. They translate the user query to a SQL statement to be executed on the product database. The query terms are mapped to predicates on the table attributes. This approach is able to tackle queries like 'small IBM laptop' with clear meaning (map on the size and brand attributes). However complex concepts i.e. 'business' for which the meaning is rather ambiguous, are associated with a textual predicate ('contains') over attributes like the product name or description. Again this approach suffers from *incompleteness* and *impreciseness*.

Our work is also related to the field of product feature extraction. In this context, Hu and Liu [20], introduce a method for considering product features implied through adjectives like 'heavy', or 'big'. For this purpose, they use a human labeled training set, and generate rules with association rule mining for the features and adjective mappings. As in the case of approaches translating the user query to SQL, this method is only feasible for queries where a clear-cut mapping between the query and table attributes can be performed. This is not the case for conceptual queries.

Turning to the field of concept extraction, in [21], Weld Hoffman and Wu propose Kylin, a self-supervised open information extraction technique. Kylin relies on information from Wikipedia to learn extractors for concepts. Wikipedia is only used as a seed, with the extractors being learned by means of bootstrapping on the Web and with the support of WordNet providing for the semantic term relations. But the extracted concepts are rather general and cannot cope with the closed vocabulary of product descriptions.

VI. CONCLUSIONS & FUTURE WORK

In this paper we presented a novel approach for supporting product search on conceptual features, combining structured product data with natural-language product reviews. Starting from the AOL query log we identified concepts as an essential building block for feature based product evaluation. The major problem with using classical retrieval approaches for this task is that user reviews mostly do not mention these features explicitly, but only hint at them in more or less semantically related terms.

Starting from a small set of comprehensive editorial reviews our novel self-learning-based approach allows identifying implicit conceptual features even in short pieces of unstructured data like e.g. user reviews. In our evaluation against classical IR baselines (VSM with TF-IDF and LSI) we have shown that our system definitely achieves superior results and can even deal with overlapping concepts. Indeed our approach outperforms classical IR methods especially for high recall values up to 90% where on average a precision of still over 50% has been proven.

Although for the discussion in this paper we only applied our approach to the restricted domain of cell phones, the results should be applicable in other product domains, too. We are currently in the process of experimenting with other product fields like laptops and cars to get a better intuition about the specific needs and the existence of conceptual features in different domains. Moreover, we also want to address conceptual queries with lower *consensus* over the result e.g. 'special', 'beautiful', 'best'. Such concepts have to be answered in a more personalized (or user profile-based) fashion.

REFERENCES

[1] D. Felix, C. Niederberger, P. Steiger, and M. Stolze, "Feature-oriented vs. needs-oriented product access for non-expert-online shoppers" in Towards the E-Society: E-Commerce, E-Business, and E-Government: Proceedings of the 1st IFIP Conference on E-Commerce, E-Business, E-Government (I3E 2001). IFIP International Federation for Information Processing, Vol. 74, pp. 399–406, Kluwer. 2001.

[2] K. Bischoff, C. Firan, W. Nejdl, and R. Paiu, "Can all tags be used for search?". Proc. of the 17th ACM conference on Information and knowledge management (CIKM'08), Napa Valley, California, USA. 2008.

[3] G. Salton, A. Wong, and C. Yang, "A vector space model for automatic indexing.", Communications of the ACM, Vol. 18(11), pp. 613–620. 1975.

[4] S. Deerwester, S. Dumais, G.W. Furnas, T.K. Landauer, and R. Harshman, "Indexing by latent semantic analysis". Journal of the American Society for Information Science, Vol. 41(6), pp. 391–407. 1990.

[5] P. Cimiano, A. Schultz, S. Sizov, P. Sorg, and S. Staab, "Explicit versus latent concept models for cross-language information retrieval". Proc. of the 21st international jont conference on artifical intelligence (IJCAI'09), pp. 1513–1518. 2009.

[6] E. M. Voorhees, "Query expansion using lexical-semantic relations". Proc. of the 17th annual international ACM SIGIR conference on research and development in information retrieval (SIGIR '94). pp 61–69. Springer-Verlag. 1994.

[7] P. Srinivasan, "Query expansion and MEDLINE". Information processing and management, Vol. 32(4), pp. 431–443, Elsevier. 1996.

[8] Y. Qiu, and H. Frei, "Concept based query expansion". Proc. of the 16th annual international ACM SIGIR conference on research and development in information retrieval (SIGIR '93), pp. 160–169, ACM Press, New York. 1993.

[9] J. F. Engel, R. D. Blackwell, and P. W. Miniard. Consumer Behaviour, 6th edition, Dryden Press, Chicago, Illinois, USA. 1990.

[10] J. Rowley, "Product search in e-shopping: a review and research propositions". Journal of Consumer Marketing, Vol. 17(1), pp. 20–35. 2000.

[11] J. Selke, S. Homoceanu, W. T. Balke, "Conceptual Views for Entity-Centric Search: Turning Data into Meaningful Concepts". To appear in Computer Science: Research and Development. DOI: 10.1007/s00450-011-0179-8. 2011.

[12] G. L. Murphy, The Big Book of Concepts, MIT Press. 2002.

[13] X. Zhou, J. Gaugaz, W. T. Balke, and W. Nejdl, "Query relaxation using malleable schemas". ACM SIGMOD International Conference on Management of Data (SIGMOD'07), pp. 545–556, Beijing. 2007.

[14] S. Agrawal, K. Chakrabarti, S. Chaudhuri, V. Ganti, C. König, and D. Xin, "Exploiting web search engines to search structured databases".

Proc. of the 18th international conference on World wide web (WWW'09) ACM, New York, NY, USA, pp. 501–510. 2009.

[15] N. Zaiqing, W. Ji-Rong, and M. Wei-Ying, "Object-level vertical search". Third Biennial Conference on Innovative Data Systems Research (CIDR'07), pp. 235–246. 2007.

[16] V. Ganti, Y. He, and D. Xin, "Keyword++: A framework to improve keyword search over entity databases". Proc. of VLDB Endow. Vol. 3, pp. 711–722. 2010.

[17] N. Sarkas, S. Paparizos, and P. Tsaparas, "Structured annotations of web queries". Proc. of the 2010 international conference on Management of data (SIGMOD'10). ACM, New York, NY, USA, pp. 771–782. 2010.

[18] C. Fellbaum, WordNet: An Electronic Lexical Database. The MIT Press. 1998.

[19] M. Hu, and B. Liu, "Mining Opinion Features in Customer Reviews". Proceedings of the 19th national conference on Artifical intelligence (AAAI'04) , pp. 755–760, AAAI Press. 2004.

[20] B. Liu, M. Hu, and J. Cheng, "Opinion Observer: Analyzing and Comparing Opinions on the Web". Proc. of the 14th international conference on World Wide Web (WWW '05). ACM, New York, NY, USA, pp. 342–351. 2005.

[21] D. Weld, R. Hoffmann, and F. Wu, "Using Wikipedia to Bootstrap Open Information Extraction". SIGMOD Rec. Vol. 37(4), pp. 62–68. 2008.

[22] C. P. Fung, J. Yu, H. Lu, and P. Yu, "Text Classification without Negative Examples Revisit". IEEE Transactions on Knowledge and Data Engineering (TKDE'06). Vol. 18(1), pp. 6–20. 2006.

[23] H. Guo, H. Zhu, Z. Guo, X. Zhang, and Z. Su, "Product feature categorization with multilevel latent semantic association". Proceeding of the 18th ACM conference on Information and knowledge management (CIKM '09). ACM, New York, NY, USA, 1087-1096. 2009.

[24] G. Salton, and C. Buckley, "Term-weighting approaches in automatic text retrieval". Information Processing and Management, pp. 513–523, Pergamon Press, Inc. 1988.

[25] V. Raghavan, P. Bollmann, and G. S. Jung, "A critical investigation of recall and precision as measures of retrieval system performance". ACM Transactions on Information Systems, Vol. 7, pp. 205–229. 1989.

[26] C. Peters, M. Braschler, J. Gonzalo, and M. Kluck, Comparative Evaluation of Multilingual Information Access Systems: 4th Workshop of the Cross-Language Evaluation Forum, CLEF 2003, Trondheim, Norway, August 21-22, 2003, Revised Papers (Lecture Notes in Computer Science), Springer-Verlag Berlin. 2004.