Probabilistic Modeling of Vague Conditions in Interactive Product Search

Alfred Sliwa

University of Duisburg-Essen, Duisburg, Germany sliwa@is.inf.uni-due.de

Abstract. Today's e-commerce platforms provide a huge amount of products attached with a rich set of properties. Modern online shops support users in the product search process by offering facet retrieval, i.e., limiting search results by using several filters. However, many systems support retrieval of items fulfilling Boolean conditions so that only exact matching leads to the inclusion in the result list. On the other side, users often have vague conditions in mind, e.g, "high resolution laptop", and cannot describe their information need precisely. Users are forced to specify crisp conditions to get any results but this leads to the problem that only a few items are returned by the system. Often, there exist other products that closely satisfy the user's information need. In this work, we present the obvious search problems with vague conditions and potential research questions w.r.t product search. We argue that leveraging probabilistic modeling techniques and especially user-driven UI development can optimize the overall retrieval quality.

Keywords: Product Search · Interactive IR · Probabilistic IR

1 Motivation

One important application in Information Retrieval (IR) is product search on ecommerce platforms. In recent years online shopping gained more popularity and the online retail worldwide increased rapidly¹. This shows that online shopping plays a significant role in today's goods acquisition and overtakes traditional shopping services.

Users frequently use online-shops either for known item search or for search based on product properties description. While the former one delivers satisfactory results, the latter often yields insufficient answer sets. This problem occurs especially in case of vague search conditions which cannot be handled by existing systems. For example, someone is searching for a "powerful and lightweight laptop with a long battery runtime for less than \$800". For such query examples, many IR systems offer full-text search, which is based on lexical comparisons and

Copyright © 2019 for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0). FDIA 2019, 17-18 July 2019, Milan, Italy.

¹ https://www.statista.com/topics/871/online-shopping/

ignore the query semantics. Phrases like "long battery runtime" would hardly lead to appropriate result items and the asking price would be interpreted as a search for the number 800 in the product description. Faceted filters are provided to assist the user in the search process by restricting search results according to specific conditions, e.g., price or display size, price value range. But in this way the user is forced to concrete her information need and to enter crisp conditions for facet filters. In some cases this is difficult; for example when a user searches for "lightweight laptops", one has to determine the threshold for "lightweight". For some product properties there are no filters available, e.g., "powerful laptops". Another issue occurs when some criteria are in conflict with each other, e.g., "powerful" vs. "cheap". By default the query conditions are strictly interpreted and must all be fulfilled like in Boolean retrieval systems. These systems reach their limits when the query becomes more complex and results in phenomena like the empty answer problem. Users have to re-formulate the query by relaxing certain conditions. But instead of this heavy-going re-formulation process, the system could help the user, if query conditions were understood as vague, and a discrimination between relevant and less-relevant criteria would be possible. In the interest of the user, it could be useful to retrieve laptops that slightly miss the query conditions. By incorporating user's query constraints, i.e., conditions that must be met, and query preferences, i.e., optional and less-relevant conditions, the IR system can distinguish among the different criteria and return more potentially relevant items.

The focus of our research lies in developing interactive IR methods that are capable of finding objects that also closely match the query conditions in the use case of product search. The concept of vagueness will be investigated with regard to text and fact conditions. We are working on the implementation of a search engine that addresses the challenges in the aforementioned problems.

2 Background and Related Work

Probabilistic IR models are based on the probabilistic ranking principle (PRP) claiming that documents should be ranked according to their probabilistic relevance w.r.t. query [9]. The treatment of vague queries considered probabilistic modeling approaches [2]. In [3] probabilistic models are used to integrate text and fact retrieval where both text and fact conditions are treated similarly and vague but separately. Probabilistic indexing weights are computed for each query condition w.r.t. an object in order to compute a final retrieval score. There are systems that try to handle vague queries in databases; for instance [7] proposes the usage of the vector space model to compare the distance between a vague query and database objects. Fuzzy logic has also been investigated in the context of vague queries and imprecise data in databases [8, 12]. While fuzzy logic – a generalization of Boolean logic – aims to compute a value similar to object's relevance degree w.r.t. a query, probabilistic logic estimates the probability of an object being relevant to a query. One advantage of the latter approach, is

that it is able to incorporate empirical data from an application for improving system performance.

In the recent past, a quantum logic based approach (QQL) was introduced by [10] explaining how query results returned by such a retrieval system could be interpreted as probability relevance scores. The basic idea is the usage of a vector space model from quantum mechanics and logic. Query and data objects are regarded as elements of this vector space model. The cosine similarity between query space vector and object vector is used to compute the probabilistic relevance score of the data object w.r.t query. Based on this theoretical quantum model [6] developed a calculus query language (CQQL) which enhances the query language by the treatment of similar predicates and query weighting. In CQQL uncertain results occur due to vague query formulations while the queried objects are regarded as certain.

3 Research Questions

In the following we describe the challenges and research questions according to vague text and fact conditions in product search by using the laptop purchase as on-going example.

Vague IR System

Vague Fact Conditions A vague query fact condition describes the extent of fulfillment according to an attribute value of one laptop. Instead of binary weights, the associated score is probabilistic and ranges from 0 to 1. There are different types of vague fact conditions to distinguish:

- Value Equality refers to a concrete value of an attribute, e.g., "display size
 14 inch"
- **Threshold** refers to a concrete value of an attribute which should not be below or above, e.g., "price < \$500"
- **Tendency** refers to a preference direction, e.g., "lightweight"

Attributes can have different scales, e.g., nominal, ordinal or metric. Depending on the vague fact condition type and the attribute scale, different approaches have to be investigated for computing a respective vagueness score. For our use case vague fact conditions can be applied to all technical product properties, e.g., CPU, memory, hard drive, display size, battery runtime, color, operating system, etc., as well as to the product price. Another obstacle is to find laptops that most closely meet all technical requirements, but for the lowest price. For each laptop an overall probabilistic relevance score w.r.t. the user query conditions can be computed. In this way, the result list includes also items with relative high probabilistic scores and not only items perfectly fitting to the user query.

Vague Text Conditions One problem in full-text search arises with mismatching vocabularies between query and document. The same concepts can be described by different words and language styles, thus making the matching procedure more complicated. When a user searches for "high performance laptops", matching on the lexical level would fail in case of laptops containing synonyms of the query phrase like "high quality notebooks" or "powerful computers". Instead of matching both sides by word-by-word comparisons which is prevalent in many search engines, we aim to develop a semantic approach capable in measuring the similarity between query and document. Previous work [1,4] proposed unsupervised methods like latent semantic analysis (LSA) to map query and document texts to low-dimensional semantic vector space in order to combat the drawbacks of lexical matching methods. Current state-of-the art systems [5, 11 aim to improve the system performance by incorporating clickthrough data to differentiate relevant documents from non-relevant ones and so representing a weakly-supervised approach. Furthermore, the utility of word hashing allows effective handling of large vocabularies by reducing the bag of word (BoW) vector space dimensionality.

Weighting and Ranking

Faceted retrieval conjunctively combines query conditions specified by the user. In order to receive any results, users often specify conditions carefully, otherwise the description of the optimal result would yield an empty result list. An alternative solution would be to introduce optional query conditions. It should be possible to weigh selected query criteria differently. Especially for binary laptop attributes it would be important to allow for optional condition selection. Moreover, the relative importance among single conditions enables a better treatment of conflicting criteria, e.g., price vs. performance. As search queries are extended by vague text and fact conditions, it is important to develop ranking functions which take vagueness as well as explicit user weighting into account.

The association between vagueness and weighting can be explained by an easy example. A user searches for a laptop with 16" display size and a SSD with 256 GB for a low price. If the user prioritizes the price over technical requirements, then the top ranked laptop in the result list could be a laptop with only 15" display size and a SSD in the desired size for \$750. A laptop fulfilling all technical criteria but being \$100 more expensive would be ranked on the second position. Only due to the query expansion by the vague fact condition w.r.t. display size, the user receives the first result.

Text to Fact Condition Mapping

Some phrases in the user query refer to property values of one laptop, e.g., "high performance notebook" refers to the attribute CPU speed. Nowadays, many IR systems do not differentiate text from fact conditions and treat the previous example query phrases as text conditions, i.e., the system retrieves only products

that contain these terms in the product title or description. But a mapping from text to fact conditions is not performed.

One possible solution to this problem is based on the analysis of query logs. Frequent phrases can be detected and mapped to their respective vague fact condition, e.g., "cheap" \rightarrow "low price". Another approach is to measure the correlation between phrases in laptop descriptions or titles with attribute values, i.e., generate rules by regarding the association of a term-attribute pair. The final performance of the transformation rules will be evaluated in real search sessions by leveraging user's implicit feedback.

Transparent Result Set Visualization

Another interesting research topic is result set visualization. One challenging task is the determination of the ordering of result objects including the snippet and overview information. Furthermore, the user should be able to interpret and understand the results returned by the system. Hence, it is necessary to implement a "White-Box" system that contains for each result item additional information about which query conditions are met and which not. The transparent result set visualization could help the user in her decision-making process. Additionally, it is useful to investigate on alternative result list representations where the user can easily read and compare conflicting conditions, e.g., "low price vs. high performance", and to select a reasonable trade-off. The interaction with the result list enables the collection of implicit feedback, e.g., relevance feedback by accepting and denying laptops. If a user accepts one laptop, the corresponding attribute values contribute to adjusting parameters of the vagueness and ranking methods.

Experimental Setting

As experimental data we crawled more than 20,000 laptops from the Amazon platform. On the one hand, we can use this dataset to model different vagueness functions by utilizing probabilistic logic like proposed in [2]. One the other hand, it can be used to compare the performance of different systems, i.e., vague system vs. boolean system. A useful evaluation experiment could be a user study where participants should solve a task with one of the two systems and judge the interaction process afterwards.

The development of the UI considers user interaction. We want to investigate how a user can affect the parameters of the different concepts, e.g., vagueness and weighting functions of fact conditions or term selection for query expansion for treatment of vague text conditions. This user feedback is relevant to judge the quality of the internal functions.

References

- Deerwester, Scott, et al. "Indexing by latent semantic analysis." Journal of the American society for information science 41.6 (1990): 391-407.
- Fuhr, Norbert. "A probabilistic framework for vague queries and imprecise information in databases." Proceedings of the 16th International Conference on Very Large Databases. Morgan Kaufman, Los Altos, California, 1990.
- 3. Fuhr, Norbert. "Integration of probabilistic fact and text retrieval." Proceedings of the 15th annual international ACM SIGIR conference on Research and development in information retrieval. ACM, 1992.
- 4. Hofmann, Thomas. "Probabilistic latent semantic indexing." ACM SIGIR Forum. Vol. 51. No. 2. ACM, 2017.
- Huang, Po-Sen, et al. "Learning deep structured semantic models for web search using clickthrough data." Proceedings of the 22nd ACM international conference on Information & Knowledge Management. ACM, 2013.
- Lehrack, Sebastian, Ingo Schmitt, and Sascha Saretz. "CQQL: A Quantum Logic-Based Extension of the Relation Domain Calculus." Proceedings of the International Workshop Logic in Databases (LID09). 2009.
- Motro, Amihai. "VAGUE: A user interface to relational databases that permits vague queries." ACM Transactions on Information Systems (TOIS) 6.3 (1988): 187-214.
- 8. Prade, Henri, and Claudette Testemale. "Generalizing database relational algebra for the treatment of incomplete or uncertain information and vague queries." Information sciences 34.2 (1984): 115-143.
- 9. Robertson, Stephen E. "The probability ranking principle in IR." Journal of documentation 33.4 (1977): 294-304.
- Schmitt, Ingo. "Qql: A db&ir query language." The VLDB JournalThe International Journal on Very Large Data Bases 17.1 (2008): 39-56.
- 11. Shen, Yelong, et al. "A latent semantic model with convolutional-pooling structure for information retrieval." Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management. ACM, 2014.
- 12. Zemankova, Maria, and Abraham Kandel. "Implementing imprecision in information systems." Information Sciences 37.1-3 (1985): 107-141.