Demonstrating Requirement Search on a University Degree Search Application

Nicholas Mendez
The University of the West Indies
St. Augustine, Trinidad and Tobago
nicholas.mendez@my.uwi.edu

Kyle De Freitas
The University of the West Indies
St. Augustine, Trinidad and Tobago
kyle.defreitas@sta.uwi.edu

Inzamam Rahaman
The University of the West Indies
St. Augustine, Trinidad and Tobago
inzamam.rahaman@my.uwi.edu

ABSTRACT

In many domains of information retrieval, we are required to retrieve documents that describe requirements on a predefined set of terms. A requirement is a relationship between a set of terms and the document. As requirements become more complex by catering for optional, alternative, and combinations of terms, efficiently retrieving documents becomes more challenging due to the exponential size of the search space. In this paper, we propose RevBoMIR, which utilizes a modified Boolean Model for Information Retrieval to retrieve requirements-based documents without sacrificing the expressiveness of requirements. Our proposed approach is particularly useful in domains where documents embed criteria that can be satisfied by mandatory, alternative or disqualifying terms to determine its retrieval. Finally, we present a graph model for representing document requirements, and demonstrate Requirement Search via a university degree search application.

CCS CONCEPTS

• **Information systems** → **Document representation**; **Document structure**; *Web applications*; Web searching and information discovery; • **Software and its engineering** → *Domain specific languages*; • **Applied computing** → Digital libraries and archives.

KEYWORDS

Requirement Search, RevBoMIR, Tuplex, Requirement Graph, DSL, Boolean Model, Requirements

ACM Reference Format:
Nicholas Mendez, Kyle De Freitas, and Inzamam Rahaman. 2019. Demonstrating Requirement Search on a University Degree Search Application. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '19), July 21–25, 2019, Paris, France.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3331184.3331402

1 INTRODUCTION

In information retrieval, models that represent documents as a set of words, phrases, or terms are described as set-theoretic. Set-theoretic models include the Boolean [6], Extended Boolean [6] and Fuzzy Retrieval Models [5]. Given a user-query expressed as a Boolean expression, these set-theoretic models return a set of documents that satisfy said query.

There are domains where problems are more aptly framed as the reverse. For example, the Boolean Model can retrieve all the resumes that contain skills requested by a given job description. However, it might also be useful to know the job descriptions that match a given candidate's skills. The aforementioned models are incompatible with this use-case as the representations of documents and queries are reversed. In RevBoMIR, we represent documents as Boolean expressions to facilitate a querying procedure in which the query is a set of terms. In applying RevBoMIR we can perform Requirement Search. We compare Requirement Search and the Boolean Model succinctly in Table 1, where T is the set of terms.

Component	Boolean Model	Requirement Search Model
Document	Any sub set of T $D = \{t_i \ldots t_j\}$	Boolean expression $D = (t_1 \wedge t_1) \vee t_2$
Query	Boolean expression $Q = (t_1 \wedge t_1) \vee t_2$	Any sub set of T $Q = \{t_i \ldots t_j\}$
Example	retrieves resumes from a job description	retrieves job descriptions for a set of skills

Table 1: Boolean vs Requirement Search Models

2 REQUIREMENT SEARCH IN EXISTING AND SIMILAR SYSTEMS

As aforementioned, Requirement Search is closest to the Boolean retrieval model. There have been numerous extensions to the Boolean model, such as Bordonga et al.'s [1] work on modelling weight terms to facilitate search result ranking. To our knowledge, no other work suggests reversing the model to fulfill a different information need.

There are also several instances in the literature where researchers have tackled problems that can be framed in terms of Requirement Searching. Yi et al. [8] presented work on matching resumes to jobs. They required resumes to be submitted into a semi-structured format and relied on relevance models to rank the retrieved resumes. Job seekers can be modelled as a collection of skills, and job descriptions require different combinations of these skills. In terms of our model, job seekers can be considered as queries, and job descriptions can be considered as documents. Moreover, we can even rank retrieved descriptions based on a suitability metric that considers minimum and optional requirements.

Online dating can also serve as a use-case for our model. Diaz et al. [2] developed a retrieval system that matched users based on

ACM ISBN 978-1-4503-6172-9/19/07…$15.00
https://doi.org/10.1145/3331184.3331402

Degree	Requirements
Computer Engineering	Mathematics and Physics
Computer Science	Mathematics AND English any 1 Science
Biochemistry	Chemistry and Biology OR either one and another Science

Table 2: Degrees and Requirements

Symbols	Requirements
M	Mathematics
P	Physics
C	Chemistry
B	Biology

Table 3: Symbols Key for Table 4

their stated preferences. A user's profile would contain a combination of preferences and it's attributes would match with other user profiles. Hence, under our model, a user's attributes would constitute a query, and a profile would constitute a document.

Suppose that you have ingredients in your pantry and would like to determine which foods you can cook using those ingredients. As shown by Kuo et al. [4], this can be seen as an information retrieval problem. From the perspective of our model, the ingredients can be considered a query, and a recipe can be considered a document. Hence, our model can be used in recipe retrieval.

Ge et al. [3] presents a food recommender which makes personal suggestions based on tags specified by the user. These tags act as terms because they help determine the retrieval of the recipe. The specification of tags is a representation of requirements.

3 PROBLEM DEFINITION

In this demonstration, we propose reversing the Boolean Model (RevBoMIR) for retrieving information where documents are conceived as Boolean expression, and a user's query is conceived as a set of terms. We refer to a document's Boolean Expression as its requirements. Requirement Search retrieves all documents that match a given set of terms such that the documents' requirements are met by the terms in the query. Representing documents as Boolean expressions in Disjunctive Normalized Form (DNF), as illustrated in Table 4, shifts the expressiveness from the user's query to the documents. Combinations of terms, alternative terms and restrictions on terms can be modeled as a document's requirements. As the document's requirements become more complex, the task of retrieving matching documents for a user's query of an arbitrary set of terms becomes more challenging. The challenge of retrieving documents is due to the large search space in matching each document to the query. Hence, there's a need to succinctly represent document requirements for efficient computation. As requirements are defined by the user, its specification should be human readable. We provide a means for users to express requirements that can be converted into a document's Boolean Expression and an implementation of Requirement-Search in the form of a web application.

Degree	Boolean Expression
Computer Engineering	$(M \wedge P)$
Computer Science	$M \wedge ((C \wedge B) \vee (B \wedge P) \vee (P \vee C))$
Biochemistry	$(C \wedge B) \vee ((C \wedge P) \vee (B \wedge P))$

Table 4: Degrees as Boolean Expressions (Symbols listed in Table 3)

4 DOMAIN-SPECIFIC LANGUAGE DESIGN

DSLs enable end-users to express concepts from their domains in a machine-readable form regardless of their programming proficiency [7]. In this section, we propose a simple DSL Tuplex, that would enable a domain-independent way to express document-requirement mappings. In our DSL, every document can be described by a top-level requirement.

4.1 Requirement Specification

A requirement in Tuplex can be described recursively as a tuple of the form (n, L) where $n \in \mathbb{Z} - \{0\}$ and L is a set that can contain terms or other requirements. Note that the absolute value of n must be less than or equal to the number of elements in L. We let a requirement where L is comprised only of terms be a base requirement. Other requirements are called complex requirements. A requirement is considered valid only if it is either a base requirement or recurses down into base requirements. When $n > 0$, a query must satisfy at least n of the contents of L. However, if $n < 0$, a query must satisfy at most $|n| - 1$ of the contents of L. For example

$$\text{Document Q} = A \cdot B + (2, \{C, D, E\}) \tag{1}$$

is the Tuplex statement for a document Q.

Note that every requirement can be expressed as a DNF Boolean expression. We outline a proof of this below:

Proof. *Base Case*: In the case of base requirements we need to consider two cases: $n > 0$ and $n < 0$.

Case 1: $n > 0$ We consider every possible combination of size n of the set L. In every such combination, we apply conjunction to every element. Since conjunction is commutative, the order of the elements in each combination is irrelevant. Each of these conjunctions form a Boolean product. Each of these products are then combined into a simple disjunction. This disjunction forms a Boolean sum. Hence, every base requirement with $n > 0$ can be expressed as a DNF Boolean expression.

Case 2: $n < 0$ We generate the same DNF as $(-n, L)$ according to case 1, negate the expression, and then apply De Morgan's law. From this expression, we can expand and simplify to yield a DNF Boolean expression. Hence, every base requirement with $n < 0$ can be expressed as a DNF Boolean expression.

Inductive Step: Suppose that we have a set of terms and requirements where every requirement in that set can be expressed as a DNF Boolean expression.

Let a complex requirement be of the form (n, L)

Case 3: $n > 0$

We can replace every requirement in L with its DNF Boolean expression. Following a similar procedure as in case 1, we get all

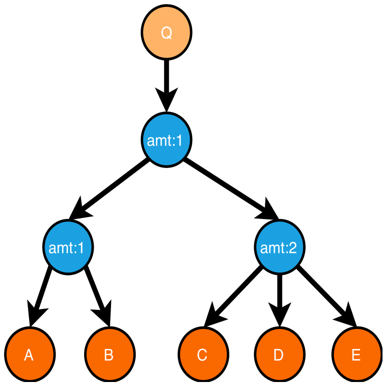

Figure 1: Graph Representation of Equation (1)

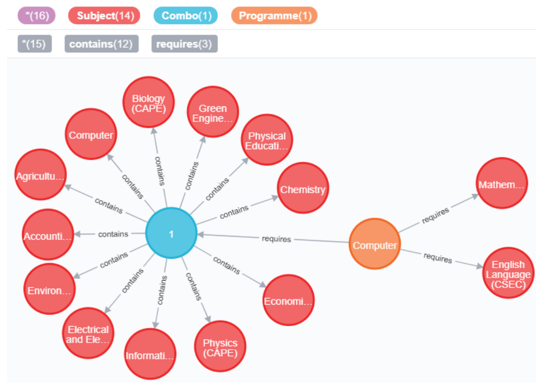

Figure 2: Computer Science Requirements

combinations of size n including the terms and these expanded Boolean expressions. We then perform conjunction as before and exploit the distributive property of the disjunction boolean operator to expand these combinations into DNF Boolean expressions. We can then apply disjunction to each of these expressions to get another DNF Boolean expression representing the entire complex requirement.

Case 4: $n < 0$

We generate the same DNF as $(-n, L)$ according to case 3, negate the expression, and then apply De Morgan's law. We then expand and simplify accordingly to get the DNF Boolean expression.

Hence, every Tuplex requirement can be expressed as a DNF Boolean expression.

□

5 GRAPH-BASED REPRESENTATION OF REQUIREMENTS

Requirement sets are modelled as a heterogeneous directed graph with the following node types:

- Document Nodes
- Combo Nodes
- Term Nodes

Combo nodes have an attribute called 'amt' that represents the integer value n in the tuple structure (n, L). In addition, Combo nodes are connected by outgoing edges to either term nodes if L is a set of terms or a another combo node if L comprises child requirements. For example, Figure 1 represents the requirements encoded by Equation (1) as a graph.

A document is directly connected to its mandatory terms. In addition, documents are connected to combo nodes for combinatorical requirements. Term nodes are never repeated; therefore different documents would have vertices to a common term node which they require. New document and term nodes may be added to the system without modifying other pre-existing document and term nodes.

6 UNIVERSITY DEGREE APPLICATION

Our initial work was based on the need to represent university degree data to be filtered based on a candidate's O-level and A-level subjects. Candidates can qualify for a degree with different combinations of subjects, as some subjects may be mandatory or optional. Due to the resulting complexity, there is a need for prospective students to quickly view all possible degree options for their respective set of subjects. The application provided interfaces for defining degree in the system by authorized parties, and for searching degrees by students. An example of the degree requirements is given in Table 2.

6.1 Cypher Query

The model was implemented on a Neo4j database, and allows users to search for all applicable programs for an arbitrary list of subjects. A query to perform this search is shown in Listing 1. Note that lines 4 and 5 of the query are validation steps relevant to the specific domain and not Requirement Search. Consequently, it is trivial to modify the query to accommodate another suitable domain.

```
1  MATCH (s:Subject), (p:Programme)
2  WHERE s.name in ['Communication Studies (CAPE)', 'Chemistry
       (CAPE)', 'Physics (CAPE)', 'Mathematics (CSEC)', '
       English Language (CSEC)', 'Physics (CSEC)']
3  WITH collect(s) as subs, p
4  WITH p, subs, SIZE(FILTER(c in subs WHERE c.level ="CSEC"))
       as csecs, SIZE(FILTER(c in subs WHERE c.level ="CAPE"))
        as capes
5  WHERE p.csec_passes <= csecs AND p.cape_passes <= capes
6  MATCH (p:Programme)-[:requires]->(s:Subject)
7  WITH p, subs, COLLECT(s) AS mandatories WHERE ALL(n IN
       mandatories WHERE n IN subs)
8  OPTIONAL MATCH (p)-[:requires]->(c:Combo)-[:contains]->(s:
       Subject)
9  WITH p, c, subs, collect(s) as list
10 WITH p, subs, collect({amt:c.amt, set:list}) as combos
11 WHERE ALL(combo in combos where SIZE(combo.set)=0 OR combo.
       amt <= size(apoc.coll.intersection(subs, combo.set)))
       RETURN p
```

Listing 1: Example Cypher query

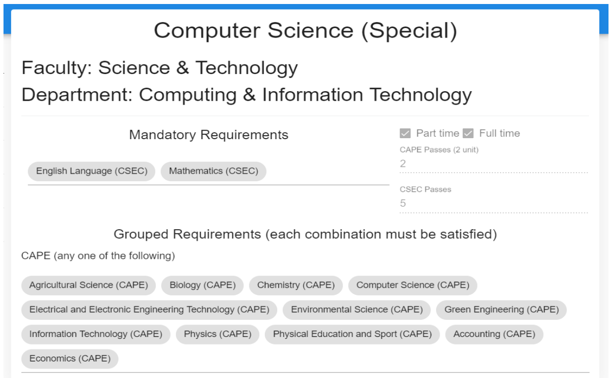

Figure 3: Computer Science Requirement Specification

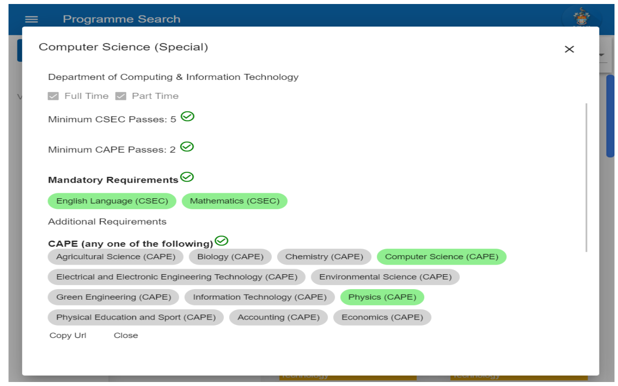

Figure 5: Degree Detail

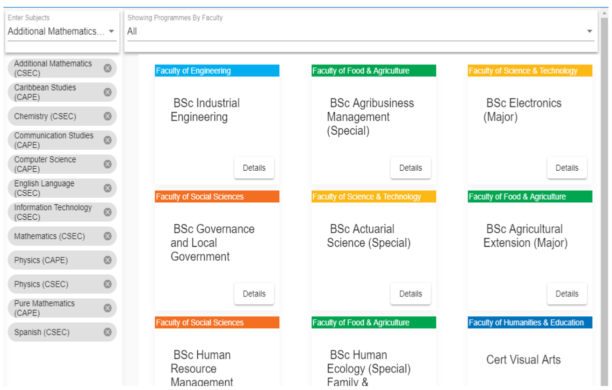

Figure 4: Requirement Search Result

7 DEMONSTRATION

7.1 Defining Degrees and Requirements

On the administrative portal, authorized users can define degree meta-data and entry requirements. For every degree, there is a field to specify mandatory components and combinatorical component to the degree's requirements. To qualify for a degree, a candidate must possess every subject in the mandatory component and at least n subjects in the combinatorical component, where different fields use a different value of n.

7.2 Requirement Search

Prospective students can select their subjects from a multi-select UI component that triggers a query against the graph database with the selected subjects. The results of applicable degrees are rendered to the right of the page. The application can be accessed at https://uwi-programme-catalog.firebaseapp.com/#/search or https://beuwi.app.

The details of the degree requirements can be verified against the user's subjects. Figure 2 shows the requirements view of the Computer Science degree which is also reflected in the graph model in Figure 1.

8 FUTURE WORK

We would like to apply our model to the domains indicated in Section 2. We also plan on benchmarking our graph model's performance against a relational model implementation. Lastly, since a requirement graph is a representation of the requirements specified in Tuplex, we can provide formal proof that all DNF Boolean expressions can be represented in a requirement graph.

9 CONCLUSION

In this paper, we have applied RevBoMIR to provide a demonstration of Requirement Search. We also provide a DSL called Tuplex and a requirements graph model to express and represent requirements respectfully.

REFERENCES

[1] Gloria Bordogna and Gabriella Pasi. 1993. A fuzzy linguistic approach generalizing Boolean Information Retrieval: A model and its evaluation. *Journal of the American Society for Information Science* 44, 2 (1993), 70–82. https://doi.org/10.1002/(SICI)1097-4571(199303)44:2<70::AID-ASI2>3.0.CO;2-I

[2] Fernando DÃŋaz, Donald Metzler, and Sihem Amer-Yahia. 2010. Relevance and ranking in online dating systems. *Proc Int ACM SIGIR Conf Res Dev Information Retrieval* 33, 66–73. https://doi.org/10.1145/1835449.1835463

[3] Mouzhi Ge, Mehdi Elahi, Ignacio Fernaández-Tobías, Francesco Ricci, and David Massimo. 2015. Using Tags and Latent Factors in a Food Recommender System. In *Proceedings of the 5th International Conference on Digital Health 2015 (DH '15)*. ACM, New York, NY, USA, 105–112. https://doi.org/10.1145/2750511.2750528

[4] Fang-Fei Kuo, Cheng-Te Li, Man-Kwan Shan, and Suh-Yin Lee. 2012. Intelligent Menu Planning: Recommending Set of Recipes by Ingredients. In *Proceedings of the ACM Multimedia 2012 Workshop on Multimedia for Cooking and Eating Activities (CEA '12)*. ACM, New York, NY, USA, 1–6. https://doi.org/10.1145/2390776.2390778

[5] Yasushi Ogawa, Tetsuya Morita, and Kiyohiko Kobayashi. 1991. A fuzzy document retrieval system using the keyword connection matrix and a learning method. *Fuzzy sets and systems* 39, 2 (1991), 163–179.

[6] Gerard Salton, Edward A Fox, and Harry Wu. 1982. *Extended Boolean information retrieval*. Technical Report. Cornell University.

[7] Arie Van Deursen, Paul Klint, and Joost Visser. 2000. Domain-specific languages: An annotated bibliography. *ACM Sigplan Notices* 35, 6 (2000), 26–36.

[8] Xing Yi, James Allan, and W. Bruce Croft. 2007. Matching Resumes and Jobs Based on Relevance Models. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '07)*. ACM, New York, NY, USA, 809–810. https://doi.org/10.1145/1277741.1277920