Diversification of search results as a fuzzy satisfiability problem

Steven Schockaert and Martine De Cock

Dept. of Applied Mathematics and Computer Science, Ghent University, Belgium
{steven.schockaert,martine.decock}@ugent.be

Abstract. In various information retrieval settings, it is of interest to the user to receive search results that are not only relevant, but also diverse. As the precise goals and the underlying understanding of diversity differs considerably from application to application, there is a need for a language in which diversification strategies can be encoded and modified in an intuitive way, yet which is sufficiently rich to capture all of the subtleties that may arise. In this paper, we propose a language based on ideas from fuzzy logics, and illustrate its flexibility and ease-of-use by providing several examples of diversification strategies. Through a number of use-case scenarios, we also point out how some of the weaknesses of existing methods can be avoided in our framework.

1 Introduction

The results returned by a search engine are useful to a user only insofar that they are relevant to her information need, are up-to-date, and arise from an authoritative source, among others. In addition to considering these qualities of individual documents, however, it is also important to ensure that the list of results is sufficiently diverse [1–3], for at least two reasons. First, when the query issued by a user is ambiguous, it makes sense to display at least one search result related to each possible understanding of the query. For example, when sending the query *apple* to google, all results on the first page are related to apple computers[1], which is cumbersome for users who are looking for information about fruit. Second, the list of search results should preferably not contain redundant results: when a given document is relevant, but very similar to a higher ranked document, it may be of little added value to the user. For example, in an image retrieval setting where the query is *Paris*, it does not make much sense to present the user with 20 photos of the Eiffel tower.

One way to deal with the problem of diversifying search results is to treat it as a combinatorial optimization problem. Starting from a given set of documents D, relevance estimates for these documents, and pairwise (dis)similarities, the primary task is then to find an optimal subset $S \subseteq D$ of k documents, which are as relevant as possible, while being as different from each other as possible. In

[1] Verified on February 14, 2011.

[3], S is selected as the set of k documents which maximizes some optimization criterion f. A first possibility is [3]:

$$f(S) = (k-1) \cdot \sum_{d \in S} \mathit{rel}(d) + 2\lambda \sum_{d_i, d_j \in S} \mathit{dist}(d_i, d_j) \tag{1}$$

where $|S| = k$, $\lambda > 0$, $\mathit{rel}(d)$ is the relevance estimate of document d and dist is a measure of dissimilarity. Two other possibilities are [3]:

$$f(S) = \min_{d \in S} \mathit{rel}(d) + \lambda \min_{d_i, d_j \in S} \mathit{dist}(d_i, d_j) \tag{2}$$

$$f(S) = \sum_{d \in S} \Big(\mathit{rel}(d) + \frac{\lambda}{|D| - 1} \sum_{d' \in D} \mathit{dist}(d, d') \Big) \tag{3}$$

Note that the latter sum ranges over the set of all documents, rather than those in S alone. As shown in [3], each of the alternatives (1)–(3) satisfies different properties, corresponding to different aspects of diversity. In general, it is not straightforward to translate a given intuition about diversification to an actual optimization criterion, which always yields the most intuitive result. Moreover, in different settings, different factors may have to be taken into account. When retrieving product reviews, for instance, it may be of interest to the user to know whether most reviews are positive or negative. In this sense, when 90% of the reviews are negative, it would not be a good idea to display 5 positive reviews and 5 negative reviews, even though this choice may maximize diversity and all 10 reviews might be relevant.

As different settings thus require different diversification mechanisms, there is a need for a flexible framework in which the intuitions underlying a particular setting can easily be translated to declarative specifications. In this paper, we propose an approach based on fuzzy logics. As in classical logic, formulas in fuzzy logics are built from constants, variables and logical connectives. In contrast to classical logic, however, formulas may take an arbitrary truth value from the unit interval $[0, 1]$ instead of only 0 (false) and 1 (true). On one hand, the resemblance with classical logic allows us to encode relations between graded properties, such as relevance or similarity, in a logical fashion. This leads to intuitive, declarative specifications, whose qualitative behavior can readily be seen from the syntactic structure of the formulas. On the other hand, Boolean connectives can be generalized to fuzzy logic connectives in different ways, which provides a form of parameterization in fuzzy logic models. The exact behavior of the resulting systems is therefore a combination of the syntactic structure of the underlying formulas, and an appropriate choice for each of the logical connectives.

The structure of this paper is as follows. In the next section, we introduce a language based on fuzzy logics, which we will use to encode diversification mechanisms. Subsequently, Section 3 presents a number of use case scenarios to illustrate some issues with existing methods such as (1)–(3). Section 4 then proposes various encodings of diversification strategies, as constraints on fuzzy logic formulas. Finally, a discussion with some concluding remarks is provided.

2 Constraints on fuzzy logic formulas

Let $D = \{d_1, ..., d_n\}$ be the set of document under consideration, with corresponding relevance scores $rel(d_i)$ and pairwise similarities $sim(d_i, d_j)$. Both relevance scores and similarities are assumed to be in $[0, 1]$. When encoding diversification mechanisms, we will also consider a number of additional predicates. If p is an m-ary predicate, then an expression of the form $p(d_{i_1}, ..., d_{i_m})$ is called a term. From these terms, formulas are constructed as follows:

- Constants in [0,1], such as $rel(d_i)$ and $sim(d_i, d_j)$, are formulas.
- Each term is a formula.
- If α and β are formulas, then $\alpha \otimes \beta$, $\alpha \oplus \beta$, $\alpha \rightarrow \beta$, $\alpha \wedge \beta$, $\alpha \vee \beta$ and $\neg\alpha$ are formulas.
- If α is a formula and $\lambda \in [0, 1]$, then $\lambda \cdot \alpha$ is a formula.
- If each of $\alpha_1, ..., \alpha_m$ are formulas, then also $avg\{\alpha_1, ..., \alpha_m\}$, $\forall\{\alpha_1, ..., \alpha_m\}$ and $\exists\{\alpha_1, ..., \alpha_m\}$ are formulas.

For the ease of presentation, we also write e.g. $\forall d \in D \,.\, p(d)$ for $\forall\{p(d) \mid d \in D\}$, or even expressions such as $avg_{d \in D}(\forall d' \in D \,.\, p(d, d'))$. There are three important differences with classical logic. First, arbitrary values from $[0, 1]$ may appear in formulas as constants. Second, there are a number of connectives that have no counterpart in classical logic, namely the scaling operator $\cdot$ and the averaging operator *avg*, which are tied to the numerical interpretation of truth degrees. Finally, there are two types of conjunction ($\wedge$ and $\otimes$) and two types of disjunction ($\vee$ and $\oplus$), which are defined as follows:

$$\alpha \wedge \beta = \min(\alpha, \beta) \qquad \alpha \vee \beta = \max(\alpha, \beta)$$
$$\alpha \otimes \beta = \max(\alpha + \beta - 1, 0) \qquad \alpha \oplus \beta = \min(\alpha + \beta, 1)$$

Note that $\wedge$ and $\otimes$ indeed correspond to logical conjunction when their arguments are restricted to the classical truth values 0 and 1, and that $\vee$ and $\oplus$ correspond to logical disjunction. The operators $\otimes$ and $\oplus$ are the connectives from Łukasiewicz logic, and provide a truth degree which is a bounded linear combination of their arguments. The operators $\rightarrow$ and $\neg$ are the implication and negation from Łukasiewicz logic, defined as

$$\alpha \rightarrow \beta = \min(1, 1 - \alpha + \beta) \qquad \neg\alpha = 1 - \alpha$$

The scaling operator $\cdot$ is simply interpreted as multiplication. Finally, *avg*, $\forall$ and $\exists$ are defined as

$$avg\{\alpha_1, ..., \alpha_m\} = \frac{\alpha_1 + ... + \alpha_m}{m}$$
$$\forall\{\alpha_1, ..., \alpha_m\} = \min(\alpha_1, ..., \alpha_m)$$
$$\exists\{\alpha_1, ..., \alpha_m\} = \max(\alpha_1, ..., \alpha_m)$$

In the following, we will consider sets E of equalities of the form $\alpha = \beta$, with α and β formulas. Such a set of equalities will be called a theory. These equalities

are seen as constraints on the possible truth values of terms, which are treated as variables. An assignment ω from terms to $[0, 1]$ is called a model of a set of equalities E if substituting every term t by its value $\omega(t)$ causes all equalities in E to be satisfied. We will consider sets of equalities E such that every model of E corresponds to a solution of the diversification problem, i.e. a reasonable choice of k documents among those in D.

By construction, all formulas can be written as the combination of a number of linear expressions using the minimum and maximum operators. Seeing terms as variables, it is therefore possible to translate a set of equalities E to a mixed integer program P, such that there is a one-on-one correspondence between the models of E and the solutions of P, using a straightforward extension of the procedure proposed in [4]. This means that models of E can be found using fast mixed integer programming solvers such as CBC[2]. Under some conditions, models can also be found using finite constraint satisfaction methods [6]. Alternatively, approximate models can be found using heuristic search techniques.

3 Motivating examples

Before illustrating how equalities of fuzzy logic formulas may be used to specify diversification mechanisms, we point out some weaknesses of existing methods using a number of scenarios:

> **Scenario A** Suppose that the set D contains two duplicates (or near-duplicates) d_1 and d_2 which are highly relevant. Ideally, only one of d_1 and d_2 should appear in the set S, no matter how relevant these documents are.

If the set S is selected based on (1) or (3), both of d_1 and d_2 may appear when these documents are sufficiently relevant and/or sufficiently different from the documents in $S \setminus \{d_1, d_2\}$ (when using (1)) or $D \setminus \{d_1, d_2\}$ (when using (3)).

Next, we consider the scenario where a query term is ambiguous, and all documents that correspond to the same understanding of the query are very similar:

> **Scenario B** Suppose that D can be partitioned in $D_1 \cup ... \cup D_m$ such that documents from the same partition are highly similar, and documents from different partitions are highly dissimilar.

Let us first assume that $m < k$. When using (1), S will then more or less be balanced, in the sense that approximately the same number of documents are chosen from each partition block D_i. However, as at least two highly similar documents will be contained in S, criterion (2) trivializes, causing many different sets S to be considered as optimal, not all of which may also be intuitively satisfactory. Finally, criterion (3) will lead to the unintuitive behavior of choosing only documents from the partition blocks with the fewest documents.

[2] `http://www.coin-or.org/projects/Cbc.xml`

Now assume that $m \geq k$. Then (1)–(2) will select one document from k different partition blocks, mainly chosen based on their relevance, while (3) would still lead to choose documents from the smallest partition blocks.

> **Scenario C** Suppose that D contains one document d_1 which is not among the k most relevant documents, but which is highly dissimilar from all other documents in D. Assume furthermore that the documents in $D \setminus \{d_1\}$ are all somewhat similar to each other.

Using (1) and (3), d_1 would typically be included in S, with the remaining documents to a large extent being chosen based on their relevance. Using (2), however, depending on the value of λ, either d_1 would not be included in S or relevance would not be taken much into account for selecting the other $k-1$ documents.

4 Encoding diversification strategies

As the previous section illustrates, it is difficult to specify global optimization criteria that always lead to those results that are intuitively most desirable. In this section, we present an alternative, in which equalities between fuzzy logic formulas encode in a declarative fashion whether a given choice of S is optimal. In particular, we introduce a predicate *imp*, such that for each document d, $imp(d)$ represents the degree to which it is important to include d in S. By construction, the set S then contains the k most important documents w.r.t. this predicate:

$$in(d) = (in_1(d) \vee ... \vee in_k(d)) \tag{4}$$

where we use $in(d)$ to denote that d is included in S and $in_i(d)$ to denote that d is the i^{th} ranked document. The terms $in_i(d)$ and $in(d)$ are assumed to be Boolean, and the right-hand side of (4) should accordingly be regarded as a Boolean expression. The following formulas encode that $in_i(d)$ should be the i^{th} most important document:

$$in_1(d) = (\forall d' \neq d \,.\, imp(d) > imp(d') \vee (imp(d) = imp(d') \wedge \neg in_1(d'))) \tag{5}$$

$$in_2(d) = (\forall d' \neq d \,.\, imp(d) > imp(d') \vee in_1(d') \vee (imp(d) = imp(d') \wedge \neg in_2(d'))) \tag{6}$$

...

$$in_k(d) = (\forall d' \neq d \,.\, imp(d) > imp(d') \vee in_1(d') \vee ... \vee in_{k-1}(d') \vee (imp(d) = imp(d') \wedge \neg in_k(d'))) \tag{7}$$

Intuitively, d should be the i^{th} ranked document if all documents which are more important are ranked higher, i.e. for every other document d' we should either have one of $in_1(d'), ..., in_{i-1}(d')$ (in which case d' is indeed ranked higher), or $imp(d) \geq imp(d')$ (in which case d is at least as important as d'). Due to the last disjunct in (5)–(7), ties are broken arbitrarily.

To complete the specification of a diversification strategy, we introduce a number of equalities to define the predicate *imp*, which together with the equalities (4)–(7) form a theory E, whose models define the optimal choices for S. As a first strategy, we may define *imp* as follows:

$$\mathit{redundant}(d) = (\exists d' \neq d \,.\, \mathit{in}(d') \wedge \mathit{sim}(d, d')) \tag{8}$$

$$\mathit{imp}(d) = \mathit{rel}(d) \otimes \neg \mathit{redundant}(d) \tag{9}$$

Note that (9) clearly reveals the intuition of the underlying diversification mechanism: it is important to include d in the set S if (i) d is relevant and (ii) no other document in S is similar to it. To conjunctively combine both aspects, the Łukasiewicz conjunction is used, which, together with the use of negation boils down to a bounded difference, i.e. $\mathit{imp}(d) = \max(0, \mathit{rel}(d) - \mathit{redundant}(d))$. The linear combination of relevance scores with redundancy scores presupposes some form of commensurability. In practice, this means that the relevance scores and similarity scores we have at our disposal may have to be manipulated somehow. Such a manipulation would moreover allow us to tweak the trade-off between relevance and similarity. Also note that (9) specifies a cyclic definition: the value of the predicate *imp* depends on the predicate *in*, which in turn depends on *imp*. The models of E thus correspond to some form of equilibria or fixpoints of these equations, an observation which can be made more explicit via the theory of fuzzy answer set programming [5]. The underlying intuition is also reminiscent of Nash equilibria, in the sense that S is defined as a set (cfr. a global strategy) which cannot be improved by replacing a single document (cfr. in which no player can improve his utility without cooperation).

Let us now reconsider the three scenarios from Section 3. In Scenario A, (9) ensures that d_1 and d_2 cannot both be included in S, as then both $\mathit{imp}(d_1)$ and $\mathit{imp}(d_2)$ would be (close to) 0. In Scenario B, assuming $m < k$, we find that at least one document from each partition block will be included in S, although the remaining documents may be chosen somewhat arbitrarily. Finally, in Scenario C, we find that d_1 would typically be included in S. Hence, in all three scenarios, more or less desirable results are found. The main problem seems to be that when selecting two highly similar documents is unavoidable, as in Scenario B, some of the remaining documents may not be selected in an optimal way. This is due to the fact that the value of $\mathit{redundant}(d)$ depends on the occurrence of a single document d' in S. In this respect, using (9) resembles the optimization criterion (2). As an alternative to (8)–(9), we may consider

$$\mathit{disparate}(d) = \mathit{avg}\{\neg \mathit{sim}(d, d') \mid d \neq d', \mathit{in}(d')\} \tag{10}$$

$$\mathit{imp}(d) = \mathit{rel}(d) \otimes \mathit{disparate}(d) \tag{11}$$

which encodes the intuition that a document d is important if, on average, the other documents in S are dissimilar to it. Using (10)–(11) in Scenario B, approximately the same number of documents will be selected from each partition block, similar as when using (1) or (3). However, in contrast to (8)–(9), using (10)–(11) does not always lead to the desired result in Scenario A. One way to

ensure optimal behavior both in Scenarios A and B would be to combine the intuitions of (9) and (11) as follows ($\lambda \in [0,1]$):

$$imp(d) = rel(d) \otimes \big(\lambda \cdot (\neg redundant(d)) \oplus (1-\lambda) \cdot disparate(d)\big) \tag{12}$$

where we assume that $\cdot$ takes priority over $\oplus$. If λ is sufficiently high, using (12) will avoid that both d_1 and d_2 are included in S in Scenario A. Moreover, in Scenario B, typically $redundant(d)$ will be close to 1 for all documents, in which case (12) behaves qualitatively similar to (11).

As already mentioned in the introduction, when ranking reviews or opinions, it is important that the set S accurately reflects whether most reviews are positive or negative, and even which type of complaints most people have (e.g. about a given product). This means that it may be beneficial to include several documents in S which express the same opinion, and are in this sense similar. To some extent, this requirement is at odds with the idea of diversifying search results, or at least, it can be seen as a tempering factor. This latter intuition of adding a tempering factor can be translated as follows:

$$prevalent(d) = avg_{d' \in D} sim(d, d') \tag{13}$$

$$imp(d) = rel(d) \otimes \big(\lambda \cdot (\neg redundant(d)) \oplus (1-\lambda) \cdot prevalent(d)\big) \tag{14}$$

which translates the intuition that d should be included if it is relevant, and it is either different from the other documents in S or it conveys a prevalent opinion. For large values of λ, (13)–(14) behave similarly as (9), while for small values of λ, diversity will only play a minimal role. To the best of our knowledge, such a trade-off has not yet been considered in existing methods.

5 Discussion

The language that was introduced in Section 2 offers the flexibility to encode a wide array of diversification strategies. In addition to the illustrations that were provided in Section 4, it is also possible to simulate existing strategies such as (1)–(3), as well as various greedy algorithms that decide which documents to add one at a time (e.g. [1]). One of the main advantages of our approach is that degrees between 0 and 1 can be treated both as numerical values (when using averaging or scaling operations), or as logical truth degrees (when using generalizations of logical connectives), which allows us to encode diversification strategies in such a way that the syntactic structure of the formulas immediately reveals the underlying intuitions.

The examples that were given correspond to basic mechanisms for diversifying search results. In practice, more structured information may be available, in which case the flexibility offered by our framework would play an even bigger role. For instance, our strategy for diversifying product reviews, i.e. (14), may be further refined when information is available about which ratings have been given by the users, or classification information about the type of complaints that are conveyed. Similarly, we may think of diversification mechanisms that

take user profiles into account, ensuring that reviews are displayed from a diverse set of users (e.g. regarding age or geographic location).

Given the observation that different applications involve different subtleties, we advocate a declarative approach, in the sense that the specification of a particular strategy should be decoupled from its implementation. One possibility for implementing the strategies encoded as constraints on fuzzy logic formulas is to translate these constraints to mixed integer programs, for which various highly efficient solvers exist. This approach has the advantage that an additional global (linear) optimization criterion can be specified to make an informed decision when there are multiple solutions. Another implementation method would be to use more heuristic techniques, e.g. taking advantage of the cyclic nature of the examples in Section 4. One idea would be to guess an arbitrary set S, i.e. a particular solution to (4)–(7), and then incrementally improve this guess by repeatedly evaluating the values of $imp(d)$, and adapting the set S accordingly.

Acknowledgments

Steven Schockaert was funded as a postdoctoral fellow of the Research Foundation – Flanders.

References

1. J. Carbonell and J. Goldstein. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st Annual International ACM SIGIR Cconference on Research and Development in Information Retrieval*, pages 335–336, 1998.
2. C. L. Clarke, M. Kolla, G. V. Cormack, O. Vechtomova, A. Ashkan, S. Büttcher, and I. MacKinnon. Novelty and diversity in information retrieval evaluation. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 659–666, 2008.
3. S. Gollapudi and A. Sharma. An axiomatic approach for result diversification. In *Proceedings of the 18th International Conference on World Wide Web*, pages 381–390, 2009.
4. R. Hähnle. Many-valued logic and mixed integer programming. *Annals of Mathematics and Artificial Intelligence*, 12:231–264, 1994.
5. J. Janssen, S. Schockaert, D. Vermeir, and M. De Cock. General fuzzy answer set programs. In *Proceedings of the 8th International Workshop on Fuzzy Logic and Applications (WILF)*, pages 352–359, 2009.
6. S. Schockaert, J. Janssen, D. Vermeir, and M. De Cock. Finite satisfiability in infinite-valued Łukasiewicz logic. In *Proceedings of the Third International Conference on Scalable Uncertainty Management*, volume 5785 of *Lecture Notes in Computer Science*, pages 240–254. 2009.