A New Lattice-Based Information Retrieval Theory

Karam Abdulahhad, Jean-Pierre Chevallet, Catherine Berrut

A New Lattice-Based Information Retrieval Theory

Karam Abdulahhad*, Jean-Pierre Chevallet**, and Catherine Berrut*

* UJF-Grenoble 1, ** UPMF-Grenoble 2, LIG laboratory, MRIM group
karam.abdulahhad,jean-pierre.chevallet,catherine.berrut@imag.fr

Abstract. Logic-based Information Retrieval (IR) models represent the retrieval decision as an implication $d \rightarrow q$ between a document d and a query q, where d and q are logical sentences. However, $d \rightarrow q$ is a binary decision, we thus need a measure to estimate the degree to which d implies q, noted $P(d \rightarrow q)$. The main problems in the logic-based IR models are the difficulties to implement the decision algorithms and to define the uncertainty measure P as a part of the logic. In this study, we chose the Propositional Logic ($\mathcal{PL}$) as the underlying framework. We propose to replace the implication $d \rightarrow q$ by the *material implication* $d \supset q$. However, we know that there is a mapping between $\mathcal{PL}$ and the lattice theory. In addition, Knuth [13] introduced the notion of *degree of inclusion* to quantify the ordering relations defined on lattices. Therefore, we position documents and queries on a lattice, where the ordering relation is equivalent to the material implication. In this case, the implication $d \rightarrow q$ is replaced by an ordering relation between documents and queries, and the uncertainty $P(d \rightarrow q)$ is redefined using the degree of inclusion measure. This new IR model is: 1- general where it is possible to instantiate most of classical IR models depending on our lattice-based model, 2- capable to formally prove the intuition of Rijsbergen about replacing $P(d \rightarrow q)$ by $P(q|d)$, and 3- easy to implement.

1 Introduction

Many studies [1–10] showed that Information Retrieval (IR) could be represented as a logical implication $d \rightarrow q$, where d represents a document and it is a set of logical sentences in a certain logic, and q represents a query and it is also a set of logical sentences in the same logic of d.

Using the logical implication $d \rightarrow q$ for representing the retrieval decision is quiet limited, because $d \rightarrow q$ is a binary decision, i.e. either d implies q or not. In addition, it is almost impossible to represent the exact semantic content of d and q [5]. We thus need a more flexible notion of implication between d and q for reflecting this loss of information. We need to estimate the degree of implication or the uncertainty of implication, noted $P(d \rightarrow q)$.

All studies that talked about representing the retrieval decision by logical implication $d \rightarrow q$, also presented some methods to estimate the uncertainty $P(d \rightarrow q)$. Most of proposals for estimating the value of $P(d \rightarrow q)$ were very

complex and costly algorithms. Even that there were a few studies, e.g. [10], presented practical and simple algorithms.

Concerning the type of logic, a wide range of logics have been used to represent d, q, and consequently $d \rightarrow q$.

Chevallet [8] uses the First-Order Logic, represented by Conceptual Graph, for representing d, q, and $d \rightarrow q$. He also uses the notion of graph projection for estimating $P(d \rightarrow q)$.

Losada et al. [10] and Abdulahhad et al. [11] use the Propositional Logic, and they use the notion of model intersection for estimating $P(d \rightarrow q)$.

Nie [3, 4] uses the Modal Logic, and he uses the notion of Possible Worlds and the relations between them for estimating $P(d \rightarrow q)$. In [4], he also uses the probability besides possible worlds for estimating $P(d \rightarrow q)$.

Meghini et al. [6] and Sebastiani [7] use the Description Logic, and they use probability to estimate $P(d \rightarrow q)$.

The choice of the appropriate logic depends on its expressive power and the complexity of its deduction algorithms, where there is a trade-off between the expressive power and deduction algorithm complexity.

In this study, we still represent d and q by logical sentences, but we depend on Lattices as an algebraic structure to position d and q and to exploit the degree of inclusion (or implication) metric defined on lattices [12, 13] for estimating $P(d \rightarrow q)$ in a simple, general, and practical way.

On the one hand, we choose the Propositional Logic ($\mathcal{PL}$), because it is simple logic and in our point of view has a sufficient expressive power to represent documents, queries, and the retrieval decision.

On the other hand, it is known that $\mathcal{PL}$ corresponds to a lattice [14]. Moreover, Knuth [12, 13] defined a degree of implication or inclusion on lattices. It is then possible to exploit the lattice structure and Knuth's notion of the degree of implication for estimating $P(d \rightarrow q)$.

This study is structured as follow: In section 2, we give a brief mathematical introduction about the lattice theory, the degree of inclusion measure, and about the propositional logic and its model-based interpretation. We claim, in section 3, that the implication $d \rightarrow q$ could be replaced by the material implication $d \supset q$. In section 4, we introduce a new mapping between the propositional logic and the lattice theory, more precisely, we introduce a new lattice where its nodes are logical clauses and its ordering relation is the material implication. In section 5, we redefine the basic IR notions depending on the lattice that defined in section 4, or in other words, we build a new IR model depending on the defined lattice and using the degree of inclusion measure for ranking. We discuss, in section 6, some important properties of our model, like its generality. We conclude in section 7.

2 Mathematical Preliminaries

2.1 Lattice

A lattice is an algebraic structure, or in other words, it is a set of elements satisfying certain properties.

Definition 2.1 (Partially Ordered Set (poset)). *A partial order over a set of elements L is a binary relation $\leq_L$, simply $\leq$, satisfying the following conditions:*

1. *Reflexivity:* $\forall a \in L, a \leq a$
2. *Antisymmetry:* $\forall a, b \in L$, *if* $a \leq b$ *and* $b \leq a$ *then* $a = b$
3. *Transitivity:* $\forall a, b, c \in L$, *if* $a \leq b$ *and* $b \leq c$ *then* $a \leq c$

The set L with the order relation $(L, \leq)$ is called a partially ordered set or poset. □

Definition 2.2 (Meet $\wedge$ and Join $\vee$). *Assume $(L, \leq)$ is a poset. For any two elements $a, b \in L$, we have:*

1. *If a unique "least upper bound" or "the supremum" of a and b exists, it is called the* ***join****, noted $a \vee b$, and it satisfies the following conditions:*
   - $a \vee b \in L$
   - $a \leq (a \vee b)$
   - $b \leq (a \vee b)$
   - $\nexists c \in L$ *where* $a \leq c$ *and* $b \leq c$ *and* $c \leq (a \vee b)$
2. *If a unique "greatest lower bound" or "the infimum" of a and b exists, it is called the* ***meet****, noted $a \wedge b$, and it satisfies the following conditions:*
   - $a \wedge b \in L$
   - $(a \wedge b) \leq a$
   - $(a \wedge b) \leq b$
   - $\nexists c \in L$ *where* $c \leq a$ *and* $c \leq b$ *and* $(a \wedge b) \leq c$

□

Definition 2.3 (Lattice). *A lattice $(L, \wedge, \vee)$ is defined either as a poset $(L, \leq)$ where the join $\vee$ and the meet $\wedge$ exist for each pair of elements in L or as an algebraic structure consisting of a set of elements L and two binary operations meet $\wedge$ and join $\vee$ satisfying:*

1. *Idempotency:* $\forall a \in L$, $a \wedge a = a$ *and* $a \vee a = a$
2. *Commutativity:* $\forall a, b \in L$, $a \wedge b = b \wedge a$ *and* $a \vee b = b \vee a$
3. *Associativity:* $\forall a, b, c \in L$, $a \wedge (b \wedge c) = (a \wedge b) \wedge c$ *and* $a \vee (b \vee c) = (a \vee b) \vee c$
4. *Absorption:* $\forall a, b \in L$, $a \wedge (a \vee b) = a \vee (a \wedge b) = a$

□

Definition 2.4 (Distributive Lattice). *Lattices that respect the following two conditions are called distributive lattices.*

1. *Distributivity of* $\wedge$ *over* $\vee$: $\forall a, b, c \in L,\ a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c)$
2. *Distributivity of* $\vee$ *over* $\wedge$: $\forall a, b, c \in L,\ a \vee (b \wedge c) = (a \vee b) \wedge (a \vee c)$

□

Definition 2.5 (Bounded Lattice). *The algebraic structure* $(L, \wedge, \vee, \top, \bot)$ *is called bounded lattice* iff $(L, \wedge, \vee)$ *is a lattice, and* $\top \in L$ *and* $\bot \in L$ *are the top and the bottom of* $(L, \wedge, \vee)$*, respectively, where:*

1. $\forall a \in L,\ a \leq \top$ *and* $a \wedge \top = a$ *and* $a \vee \top = \top$
2. $\forall a \in L,\ \bot \leq a$ *and* $a \vee \bot = a$ *and* $a \wedge \bot = \bot$

□

Definition 2.6 (Complemented Lattice). *If for any element* $a \in L$ *in the bounded lattice* $(L, \wedge, \vee, \top, \bot)$*, there exists a unique element* $b \in L$*, noted* $b = \neg a$*, satisfying:*

1. $a \wedge \neg a = \bot$
2. $a \vee \neg a = \top$

The algebraic structure $(L, \wedge, \vee, \neg, \top, \bot)$ *is called a complemented lattice.* □

Definition 2.7 (Boolean Algebra). *Any distributive and complemented lattice* $(L, \wedge, \vee, \neg, \top, \bot)$ *is a Boolean algebra.* □

Definition 2.8 (Consistency Relations). *In any lattice* $(L, \wedge, \vee)$*, the consistency relations explicitly express the relationship between the order relation* $\leq$ *and the meet* $\wedge$ *and the join* $\vee$ *binary operations, as follow:*

$$\forall a, b \in L, a \leq b \Leftrightarrow \begin{array}{l} a \wedge b = a \\ a \vee b = b \end{array}$$

□

Definition 2.9 (Sublattice). *Assume that* $(L, \wedge, \vee)$ *is a lattice.* $(L', \wedge, \vee)$ *is a sublattice of* $(L, \wedge, \vee)$ iff*,*

1. $L' \subseteq L$ *and*
2. $\forall a, b \in L',\ a \wedge b \in L'$ *and* $a \vee b \in L'$*.*

□

2.2 The Degree of Inclusion

In any poset $(O, \leq)$, it is possible and helpful to quantify the notion of inclusion by introducing the *zeta function* (2.1).

$$\forall x, y \in O, \zeta(x, y) = \begin{cases} 1 & \text{if} \quad x \leq y \\ 0 & \text{if} \quad x \not\leq y \end{cases} \tag{2.1}$$

The *zeta function* describes whether y includes x or not. It is possible to define its dual function *dual of the zeta function* (2.2).

$$\forall x, y \in O, \zeta^{\partial}(x, y) = \begin{cases} 1 & \text{if} \quad x \geq y \\ 0 & \text{if} \quad x \not\geq y \end{cases} \tag{2.2}$$

The *dual of the zeta function* describes whether x includes y or not. However, knowing that y includes x, $x \leq y$, for any two distinct elements x any y of a poset $(O, \leq)$, then clearly x does not include y, $x \not\geq y$. Even x does not include y, it is possible to describe the degree to which x includes y. Knuth [13] generalizes the inclusion (2.2) to the degree of inclusion represented by real numbers. He introduced the z *function* (2.3).

$$\forall x, y \in O, z(x, y) = \begin{cases} 1 & \text{if} \quad x \geq y \\ 0 & \text{if} \quad x \wedge y = \bot \\ z & \text{otherwise, where} \quad 0 < z < 1 \end{cases} \tag{2.3}$$

$z(x, y)$ quantifies the degree to which x includes y. Knuth [13] says: "The motivation here is that, if we are certain that x includes y then we want to indicate this knowledge. However, if we know that x does not include y, then we can quantify the *degree* to which x includes y".

Assume that instead of working with elements of poset $x, y \in O$, we work with elements of a *Distributive Lattice* $x, y \in L$ where $(L, \wedge, \vee)$ is a distributive lattice. In this case, the function z should be consistent with the structure of distributive lattices. The function z should therefore satisfy the following rules by which the degree of inclusion should be manipulated [13]: for any distributive lattice $(L, \wedge, \vee)$, $\forall x, y, t \in L$:

1. Sum rule
$$z(x \vee y, t) = z(x, t) + z(y, t) - z(x \wedge y, t) \tag{2.4}$$
2. First Product rule
$$z(x \wedge y, t) = z(x, t) + z(y, t) - z(x \vee y, t) \tag{2.5}$$
3. Second Product rule
$$z(x \wedge y, t) = C \cdot z(x, t) \cdot z(y, x \wedge t) \tag{2.6}$$
where the constant C acts as a normalization factor.
4. Bay's Theorem rule
$$z(y, x \wedge t) = \frac{z(y, t) \cdot z(x, y \wedge t)}{z(x, t)} \tag{2.7}$$

The z function is simply the probability when it is defined on a Boolean algebra or lattice $(B, \wedge, \vee, \neg, \top, \bot)$ (2.8) [12, 13].

$$\forall x, y \in B, P(x|y) = z(x, y) \tag{2.8}$$

where P is a probability function.

2.3 Propositional Logic $\mathcal{PL}$

This study depends on Propositional Logic $\mathcal{PL}$ as a theoretical and mathematical basis. Therefore, when saying that s is a logical sentence, we implicitly mean that s is a logical sentence under $\mathcal{PL}$.

Definition 2.10 (Alphabet A). *We define $A = \{a_1, \ldots, a_n\}$ as a set of all atomic propositions. The set A forms our alphabet, and it is a finite set $|A| = n$. Any proposition $a_i \in A$ can take only one of two possible values: True T, or False F.* □

A formal logic has a syntax but can also have a semantic. This semantic translates the formal sentences of that logic into another mathematical world. For example, we get the semantic of a logical sentence in $\mathcal{PL}$ by assigning a truth value (T or F) to each proposition in that sentence.

Any set of atomic propositions A corresponds to $2^{|A|}$ possible translations or interpretations Δ_A depending on the truth values assignations.

Definition 2.11 (Interpretations Δ_A). *The set of interpretations Δ_A of a set of atomic propositions A is defined as follow:*

$$\Delta_A = \{\delta_i | \delta_i = \{(a_1, x_1^i), \ldots, (a_n, x_n^i)\}, a_j \in A, x_j^i \in \{T, F\}\}$$

where $|\Delta_A| = 2^{|A|}$ because each proposition $a_i \in A$ could have one of two possible values (T or F), and $\forall \delta_i \in \Delta_A, |\delta_i| = |A|$. It is possible to build a new notation Δ'_A simpler than but equivalent to Δ_A, as follow:

$$\Delta'_A = \{\delta'_i | \delta'_i = \{a_j | (a_j, T) \in \delta_i\}\}$$

where $|\Delta'_A| = |\Delta_A|$, and δ'_i contains the propositions that have true as a truth value in δ_i. For a specific interpretation δ'_i, all propositions that do not belong to δ'_i are implicitly false. In the rest of this study we will use this simple notation for interpretations. □

In other words, for any alphabet A, the set of interpretations actually correspond to the different rows of the truth table that is built in terms of A. For example, suppose that $A = \{a, b, c\}$ contains three propositions, the truth table, the set of interpretations Δ_A, and the set Δ'_A are depicted in Figure 2.1.

Definition 2.12 (Models M). *For any logical sentence s, the subset $M_s \subseteq \Delta'_A$ that make s true is called the set of* models *of s, noted $M_s \models s$ (M_s* models *s), where: for any model $m \in M_s$, if we substitute each atomic proposition in s by its truth value in m then the truth value of s will be true.*

The notation $\models s$ means that s is a tautology, or in other words, s is true under any interpretation ($M_s = \Delta'_A$). The notation $\not\models s$ means that s is false under all interpretations ($M_s = \phi$).

We have $M_s \subseteq \Delta'_A$ is a set of models and each model $m \in M_s$ is a set of atomic propositions $m \subseteq A$. □

Fig. 2.1. The truth table of $A = \{a, b, c\}$, the corresponding set of interpretations Δ_A, and the set Δ'_A, where $\Delta_A = \{\delta_1, \delta_2, \delta_3, \delta_4, \delta_5, \delta_6, \delta_7, \delta_8\}$ and $\Delta'_A = \{\delta'_1, \delta'_2, \delta'_3, \delta'_4, \delta'_5, \delta'_6, \delta'_7, \delta'_8\}$

a	b	c	δ_i	δ'_i
F	F	F	$\delta_1 = \{(a, F), (b, F), (c, F)\}$	$\delta'_1 = \{\}$
F	F	T	$\delta_2 = \{(a, F), (b, F), (c, T)\}$	$\delta'_2 = \{c\}$
F	T	F	$\delta_3 = \{(a, F), (b, T), (c, F)\}$	$\delta'_3 = \{b\}$
F	T	T	$\delta_4 = \{(a, F), (b, T), (c, T)\}$	$\delta'_4 = \{b, c\}$
T	F	F	$\delta_5 = \{(a, T), (b, F), (c, F)\}$	$\delta'_5 = \{a\}$
T	F	T	$\delta_6 = \{(a, T), (b, F), (c, T)\}$	$\delta'_6 = \{a, c\}$
T	T	F	$\delta_7 = \{(a, T), (b, T), (c, F)\}$	$\delta'_7 = \{a, b\}$
T	T	T	$\delta_8 = \{(a, T), (b, T), (c, T)\}$	$\delta'_8 = \{a, b, c\}$

Fig. 2.2. The truth table of the material implication $\supset$, the set of interpretations Δ'_A, and the set of models $M \models a \supset b$, where $M = \{\delta'_1, \delta'_2, \delta'_4\}$

a	b	δ'_i	$a \supset b$	
F	F	$\delta'_1 = \{\}$	T	$\{\delta'_1\} \models a \supset b$
F	T	$\delta'_2 = \{b\}$	T	$\{\delta'_2\} \models a \supset b$
T	F	$\delta'_3 = \{a\}$	F	$\{\delta'_3\} \not\models a \supset b$
T	T	$\delta'_4 = \{a, b\}$	T	$\{\delta'_4\} \models a \supset b$

For example, assume that $A = \{a, b\}$ and assume that the logical sentence s is the material implication $a \supset b$. The set of models $M \models a \supset b$ is depicted in Figure 2.2, where $M = \{\delta'_1, \delta'_2, \delta'_4\}$.

Theorem 2.1 (Sentences vs. Models). *By moving from the syntax space to the semantic space, for any two logical sentences s_1 and s_2 we have:*

$$[\models s_1 \supset s_2] \Leftrightarrow [M_{s_1} \subseteq M_{s_2}]$$

where $M_{s_1} \models s_1$ is the set of models of s_1, $M_{s_2} \models s_2$ is the set of models of s_2.

Proof. $\models s_1 \supset s_2$ *means that any model of s_1 should also be a model of s_2, noted $M_{s_1} \models s_2$ or simply $s_1 \models s_2$. From the truth table in Figure 2.2, $s_1 \supset s_2$ is true* iff *there is no model of s_1 is not a model of s_2, which means $M_{s_1} \subseteq M_{s_2}$.* □

Theorem 2.2 (Models-Based Boolean Algebra B_M). *The algebraic structure $B_M = (2^{\Delta'_A}, \wedge, \vee, \neg, \top, \bot)$ is a Boolean algebra and we refer to it as Models-Based Boolean algebra, where:*

1. $2^{\Delta'_A}$ *is the power set of* Δ'_A
2. $\forall M_1, M_2 \in 2^{\Delta'_A}, M_1 \wedge M_2 = M_1 \cap M_2$
3. $\forall M_1, M_2 \in 2^{\Delta'_A}, M_1 \vee M_2 = M_1 \cup M_2$
4. $\forall M \in 2^{\Delta'_A}, \neg M = \overline{M}$ *where* $\overline{M} = \Delta'_A \setminus M$

5. $\top = \Delta'_A$
6. $\bot = \phi$

The ordering relation $\leq$ *defined on* B_M *is:*

$$\forall M_1, M_2 \in 2^{\Delta'_A}, [M_1 \leq M_2] \Leftrightarrow [M_1 \subseteq M_2]$$

□

From Theorems 2.1 and 2.2, the order relation on B_M is equivalent to the material implication. Therefore, the implication of the Boolean lattice B_M can be generalized to the degree of implication represented by real numbers (2.9).

$$\forall M_x, M_y \in \Omega_A, P(M_y|M_x) = \begin{cases} 1 & \text{if} \quad M_x \subseteq M_y \\ 0 & \text{if} \quad M_x \cap M_y = \phi \\ p & \text{otherwise, where} \quad 0 < p < 1 \end{cases} \tag{2.9}$$

where x and y are two logical sentences, $M_x \models x$ is the set of models of x, and $M_y \models y$ is the set of models of y. We know that $[\models x \supset y] \Leftrightarrow [M_x \subseteq M_y]$ (Theorem 2.1). We also know that each set of models M_x correspond to a set of logically-equivalent sentences (it is possible to choose x as a representative of this equivalent class). Therefore, for any two logical sentences x and y, $P(M_y|M_x)$ (2.9) could be rewritten as (2.10).

$$P(y|x) = \begin{cases} 1 & \text{if} \quad x \supset y \\ 0 & \text{if} \quad x \wedge y = F = \bot \\ p & \text{otherwise, where} \quad 0 < p < 1 \end{cases} \tag{2.10}$$

Here also, Knuth [13] says: "Probabilities are functions of pairs of logical statements and quantify the degree to which one logical statement implies another".

3 Revisiting IR in terms of $\mathcal{PL}$

Many studies [1–10] argue that the retrieval process could be represented as a logical implication between a document d and a query q. They say: if d and q are sets of logical sentences in a specific logic then d should be retrieved *iff* it logically implies q, noted $d \rightarrow q$. In this study, we use the Propositional Logic $\mathcal{PL}$ as underlying logic.

Rijsbergen [1] formalized the implication $\rightarrow$, as follows: for any two sentences or sets of sentences X and Y, $X \rightarrow Y$ means that '*if* X is true *then* Y'. In other words, if both X and Y are true under an interpretation then $X \rightarrow Y$ is also true under that interpretation. The truth of $X \rightarrow Y$ does not simply depend on the evaluation of X and Y in one interpretation.

Chiaramella et al. [5], by their turn, formalized the implication $\rightarrow$, as follow: $D \rightarrow Q$ is true *iff* Q is true given that D is true.

However, all studies [1–10] claim that the implication $\rightarrow$ is different from the classic material implication $\supset$. In addition, $\rightarrow$ is more appropriate than $\supset$ for IR. In this paper, we claim that the two implications $\rightarrow$ and $\supset$ are equivalent.

All definitions of the implication $X \rightarrow Y$, depicted in [1–10], agree that $X \rightarrow Y$ can only be evaluated in the cases where the antecedent X is true and in those cases the consequent Y should also be true. In other words, the evaluation space for $X \rightarrow Y$ is restricted to the cases or interpretations that make X true, whereas the evaluation space of $X \supset Y$ contains all the possible cases or interpretations.

We think that the main problem in those studies [1–10] was the inability to imagine the meaning of $d \rightarrow q$ when d is false, where d represents a document and q represents a query. However, the impossibility of imagining some thing is not a sufficient reason for do not modeling that thing. In other words, $d \rightarrow q$ when d is false is a part of the model even if this case is not realistic.

Before we go forward, we should clarify the potential map between the logical notions and the IR notions:

1. Each proposition $a_i \in A$ corresponds to only one unique indexing term. Therefore, A is also the set of indexing terms.
2. Any document d and query q are logical sentences in $\mathcal{PL}$, where: d is equivalent to a set of models $M_d \models d$, and q is equivalent to a set of models $M_q \models q$.
3. We claim that the retrieval decision $d \rightarrow q$ is equivalent to the material implication $d \supset q$ between the two sentences d and q. Theorems 2.1 and 3.1 show that:

$$[s_1 \rightarrow s_2] \Leftrightarrow [s_1 \supset s_2] \tag{3.1}$$

Theorem 3.1. *The IR implication $\rightarrow$, which is defined in [1] and [5], is equivalent to the set inclusion between models. For any two logical sentences s_1, s_2*

$$[\models s_1 \rightarrow s_2] \Leftrightarrow [M_{s_1} \subseteq M_{s_2}]$$

where $M_{s_1} \models s_1$ is the set of models of s_1, and $M_{s_2} \models s_2$ is the set of models of s_2.

Proof. *1. $s_1 \rightarrow s_2$ is true means that 'if s_1 is true then s_2', or every model of s_1 should also be a model of s_2.*
2. $M_{s_1} \subseteq M_{s_2}$ means that when s_1 is true then s_2 is also true.

□

Equation 3.1 does not contradict with previous studies, which say that queries are only evaluable in the cases (interpretations) where d is true. From the truth table in Figure 2.2 and after replacing a by d and b by q, we have:

$$M_{d \supset q} = (M_d \cap M_q) \cup \overline{M_d} \tag{3.2}$$

Although that $\overline{M_d}$ are possible models of $d \supset q$, the implementations of IR models do not, in general, take $\overline{M_d}$ into account, and they neglect them, because

$\overline{M_d}$ corresponds to the cases where d is false. Therefore, we think that modeling the cases where d is false will not pose any problem, as long as, these cases will be neglected at the implementation time.

After the previous discussion our main hypothesis is: *the material implication $d \supset q$ is the appropriate implication for modeling the retrieval decision in the IR field*. Henceforth, we will use the two implications $d \rightarrow q$ and $d \supset q$ interchangeably.

In general, IR is an uncertain process [5], so we need to evaluate the uncertainty of $d \rightarrow q$. Rijsbergen [1] and Nie [2] depend on the *Possible Worlds* semantic to evaluate the uncertainty of the implication $P(d \rightarrow q)$. More precisely, according to Rijsbergen [1] the $P(d \rightarrow q)$ could be replaced by the conditional probability $P(q|d)$, whereas, Nie [2] depends on the notion of distance between the possible worlds over the path from d to q.

In this section, we also introduce, in an intuitive manner, a way of estimating the uncertainty of an implication. We could simply claim that:

$$P(d \rightarrow q) = |M_d \cap M_q| \tag{3.3}$$

The intuitive meaning of this formula could be that the limit to which d and q are compatible, or how many system (people) could assign the same interpretation for both d and q. In next sections, we will introduce a more formal measure for estimating the implication uncertainty.

4 A New Logic-Based Lattice

The application field of this study will be the Information Retrieval (IR) field. IR could be formalized depending on the lattice of models B_M (Theorem 2.2). Any document or query could be represented as a node in B_M. However, B_M is a very general and huge lattice (if $|A| = 3$ then $|2^{\Delta'_A}| = 2^{2^3} = 2^8$). Moreover, IR notions like documents and queries are generally modelized in a very simple way e.g. set of terms. Therefore, we here propose a new lattice B_C depending on rewriting the sentences that represent documents and queries in their DNF form. This new lattice will simplify the B_M lattice, without loss of generality.

Definition 4.1 (Clauses C_A). *We define the set of clauses C_A on the alphabet A as follow:*

$$C_A = (\{a_1, \neg a_1, T\} \times \cdots \times \{a_n, \neg a_n, T\}) \setminus \{T\}$$

where $|C_A| = 3^{|A|} - 1$ and $\{a_i, \neg a_i, T\} \times \{a_j, \neg a_j, T\} = \{a_i \wedge a_j, a_i \wedge \neg a_j, a_i, \neg a_i \wedge a_j, \neg a_i \wedge \neg a_j, \neg a_i, a_j, \neg a_j, T\}$.

Note that $\{a_i, \neg a_i, T\} \times \{a_j, \neg a_j, T\} = \{a_j, \neg a_j, T\} \times \{a_i, \neg a_i, T\}$ because the conjunction $\wedge$ is commutative. Any clause $s \in C_A$ can be defined as follow:

$$\forall s \in C_A, \exists A_s \subseteq A, s = \bigwedge_{a_i \in A_s} b_i$$

where $A_s \neq \phi$ and b_i is a literal. Any literal b_i is an atomic proposition a_i or its negation $\neg a_i$. □

It is possible to split the set A_s into two disjoint sets: A_s^+ that contains the propositions occurring in their non-negative form, and A_s^- that contains the propositions occurring in their negative form, where $A_s^+ \cup A_s^- = A_s$ and $A_s^+ \cap A_s^- = \phi$. We also define the set $A_s^\pm = A \backslash A_s$, which contains the propositions that do not occur in s.

Definition 4.2 (Alphabet Splitting). *Each clause $s \in C_A$ splits the alphabet A into three mutually disjoint sets of atomic propositions:*

1. *A_s^+ contains the propositions $a_i \in A_s$ where $b_i = a_i$.*
2. *A_s^- contains the propositions $a_i \in A_s$ where $b_i = \neg a_i$.*
3. *$A_s^\pm = A \setminus A_s$ contains the propositions that do not occur in s.*

□

Example 4.1. Suppose $A = \{a, b\}$ then $C_A = (\{a, \neg a, T\} \times \{b, \neg b, T\}) \setminus \{T\} = \{a \wedge b, a \wedge \neg b, a, \neg a \wedge b, \neg a \wedge \neg b, \neg a, b, \neg b\}$. □

Example 4.2. Suppose $A = \{a, b, c\}$, for the clause $s = \neg a \wedge b \in C_A$ we have: $A_s = \{a, b\}$, $A_s^+ = \{b\}$, $A_s^- = \{a\}$, and $A_s^\pm = \{c\}$. □

Any clause $s \in C_A$ corresponds to a set of models M_s (Definition 2.12), where $M_s \models s$. In any model $m \in M_s$ of the clause s, the propositions A_s^+ should be mapped into true, the propositions A_s^- should be mapped into false, and the propositions $A_s^\pm$ could be mapped into true or false. Hence, the number of models $|M_s|$ of any clause s is $|M_s| = 2^{|A_s^\pm|}$.

For any model $m \in M_s$ and any proposition $a_i \in A$:

1. If $a_i \in A_s^+$ then a_i should be mapped into true in any model of s, so $a_i \in m$.
2. If $a_i \in A_s^-$ then a_i should be mapped into false in any model of s, so $a_i \notin m$.
3. If $a_i \in A_s^\pm$ then a_i could be mapped either into true or false in any model of s, so whether $a_i \in m$ or not.

Definition 4.3 (Full Clause Model). *Depending on the set of models M_s, we define one unique model m_s, where $\{m_s\} \models s$, as follow:*

$$\forall s \in C_A, m_s = \bigcap_{m \in M_s} (m)$$

where $M_s \models s$ is the set of models of s. m_s is the full clause model of s. □

The model m_s is equivalent to the clause s after completing it by the negation form of all propositions that do not occur in it $a_i \in A_s^\pm$. For example, suppose $A = \{a, b, c\}$ and $s = a \wedge b$ then the model m_s is equivalent to the following clause $a \wedge b \wedge \neg c$. In other words, m_s will only contain the propositions that occur in s in their non-negative form.

Theorem 4.1 (Full Clause Models Boolean Algebra B_C). *The algebraic structure $B_C = (\Psi_A, \wedge, \vee, \neg, \top, \bot)$ is a Boolean algebra and we refer to it as the Full Clause Models Boolean algebra, where:*

1. $\Psi_A = \{m_s | s \in C_A\}$
2. $\forall m_1, m_2 \in \Psi_A, m_1 \wedge m_2 = m_1 \cap m_2$
3. $\forall m_1, m_2 \in \Psi_A, m_1 \vee m_2 = m_1 \cup m_2$
4. $\forall m \in \Psi_A, \neg m = \overline{m}$ *where* $\overline{m} = A \setminus m$
5. $\top = m_\top$ *where* $m_\top = A$ *which is the full clause model of the clause* $a_1 \wedge \cdots \wedge a_n$.
6. $\bot = m_\bot$ *where* $m_\bot = \phi$ *which is the full clause model of clauses that do not contain any proposition in non-negative form, e.g.* $\neg a_i$.

The ordering relation $\leq$ *defined on* B_C *is:*

$$\forall m_1, m_2 \in \Psi_A, [m_1 \leq m_2] \Leftrightarrow [m_1 \subseteq m_2]$$

□

Theorem 4.2 (Relationship between Material Implication $\rightarrow$ and B_C). *The potential relationship between the material implication* $\rightarrow$ *and the ordering relations* $\subseteq$ *defined on* B_C *is:*

$$\forall s_1, s_2 \in C_A, [\models s_1 \rightarrow s_2] \Leftrightarrow [(m_{s_2} \subseteq m_{s_1}) \wedge (M_{s_1} \cap M_{s_2} \neq \phi)]$$

where $M_{s_1} \models s_1$ *is the set of all models of* s_1, $M_{s_2} \models s_2$ *is the set of all models of* s_2, *and* m_{s_1}, m_{s_2} *are the full clause models of* s_1 *and* s_2, *respectively.*

Proof. From the following points, it is possible to prove this theorem.

1. *From Theorem 2.1, we have:* $[\models s_1 \rightarrow s_2] \Leftrightarrow [M_{s_1} \subseteq M_{s_2}]$.
2. $[M_{s_1} \subseteq M_{s_2}] \Rightarrow [\mu(M_{s_2}) \subseteq \mu(M_{s_1})] \Rightarrow [m_{s_2} \subseteq m_{s_1}]$.
3. $[M_{s_1} \subseteq M_{s_2}] \Rightarrow [M_{s_1} \cap M_{s_2} \neq \phi]$, *this is correct knowing that* $M_{s_1} \neq \phi$ *because* $A_{s_1} \neq \phi$.
4. $[M_{s_1} \cap M_{s_2} \neq \phi] \Rightarrow \left[\left(A^+_{s_1} \cap A^-_{s_2} = \phi\right) \wedge \left(A^-_{s_1} \cap A^+_{s_2} = \phi\right)\right]$. *In other words, there is no proposition occurring in two different forms in* s_1 *and* s_2.
5. $[m_{s_2} \subseteq m_{s_1}] \Rightarrow \left[A^+_{s_2} \subseteq A^+_{s_1}\right]$. *According to the two sets* $A^-_{s_1}$ *and* $A^-_{s_2}$, *we have:*
   (a) *If* $\left(A^-_{s_2} \subseteq A^-_{s_1}\right)$ *then* $(M_{s_1} \subseteq M_{s_2})$ *because* s_1 *has more propositions and hence has less number of models. We have* $A^\pm_{s_1} \subseteq A^\pm_{s_2}$, *so* $|M_{s_1}| \leq |M_{s_2}|$.
   (b) *Otherwise* $\left(A^-_{s_1} \subset A^-_{s_2}\right)$, *we define the set* $A^-_{s_2 \setminus s_1} = A^-_{s_2} \setminus A^-_{s_1} = A^\pm_{s_1} \cap A^-_{s_2}$, *which contains the propositions that are mapped into false in* s_2 *and could be mapped into either true or false in* s_1. *As all propositions* $A^\pm_{s_1}$ *that do not occur in* s_1 *could be either true or false then it is possible to assume, without loss of generality, that:*
   1- all propositions $A^-_{s_2 \setminus s_1}$ *are false in* s_1, *or in other words, we move the set* $A^-_{s_2 \setminus s_1}$ *of propositions from* $A^\pm_{s_1}$ *to* $A^-_{s_1}$. *In this case, we have* $A^-_{s_2} = A^-_{s_1}$, *and we back to the case (a),*
   2- all propositions $A^-_{s_2 \setminus s_1}$ *are true in* s_1, *or in other words, we move the set* $A^-_{s_2 \setminus s_1}$ *of propositions from* $A^\pm_{s_1}$ *to* $A^+_{s_1}$. *In this case, we have a contradiction with point (4), and hence, this assumption is not possible. Actually, this operation of moving some propositions from one set to another will not change any thing concerning the models* m_{s_1} *and* m_{s_2} *(Definition 4.3).*

From points 1, 2 and 3, we prove that:

$$[\models s_1 \rightarrow s_2] \Rightarrow [(m_{s_2} \subseteq m_{s_1}) \wedge (M_{s_1} \cap M_{s_2} \neq \phi)]$$

From points 4 and 5, we prove that:

$$[\models s_1 \rightarrow s_2] \Leftarrow [(m_{s_2} \subseteq m_{s_1}) \wedge (M_{s_1} \cap M_{s_2} \neq \phi)]$$

□

Theorem 4.2 shows that the ordering relation $\leq$ defined on the lattice B_C is equivalent to the materiel implication $\rightarrow$ defined on the clauses of C_A (Definition 4.1).

5 Revisiting IR in terms of B_C and the z Function

As we said in previous sections, the alphabet A forms the set of terms because each term corresponds to one unique proposition, both documents and queries are logical sentences in $\mathcal{PL}$, the retrieval decision is modeled by the implication $d \rightarrow q$, and finally the degree of implication $P(d \rightarrow q)$ forms the ranking mechanism.

5.1 Documents and Queries

In IR, documents and queries could be written as a clause of C_A (Definition 4.1) or a set of clauses of C_A connected by disjunction. In other words, documents and queries are DNF logical sentences. Any logical sentence could be rewritten in the DNF form. Therefore, there are no restrictions on the logical sentences that could be used to represent documents and queries, and hence there is no loss of generality. More formally, suppose we have a set of documents D and a query q:

Definition 5.1 (Document). *Any document $d \in D$ is a DNF logical sentence, and it corresponds to one unique non-empty set of clauses $S_d \subseteq C_A$ connected by disjunction, or equivalently:*

$$\forall d \in D, d = \bigvee_{s_i \in S_d} s_i$$

where $S_d \neq \phi$. The set of models M_d of a document d is defined as follow:

$$M_d = \bigcup_{s_i \in S_d} (M_{s_i})$$

□

Definition 5.2 (Query). *The query q is a DNF logical sentence, and it corresponds to one unique non-empty set of clauses $S_q \subseteq C_A$ connected by disjunction, or equivalently:*

$$q = \bigvee_{s_i \in S_q} s_i$$

where $S_q \neq \phi$. The set of models M_q of a query q is defined as follow:

$$M_q = \bigcup_{s_i \in S_q} (M_{s_i})$$

□

5.2 Relevance

Most of studies in the logical IR models defined the relevance between a document d and a query q by an implication between them [1–10]. d is relevant concerning q *iff* d implies q, noted $d \rightarrow q$. We claim that the implication $d \rightarrow q$ could be replaced by the material implication $d \supset q$ (Equation 3.1). Of course, that is right if we use the propositional logic $\mathcal{PL}$ as an underling logic.

Assume that d is a document (Definition 5.1) and q is a query (Definition 5.2). We have $M_d \models d$ the set of document models and $M_q \models q$ the set of query models. Theorem 4.2 shows that $\models d \rightarrow q$ is equivalent to two conditions:

- **C.1**: $(\underline{M_d \cap M_q \neq \phi})$ which means that there is no contradiction between d and q, or in other words, $d \wedge q \neq F$. The verification of this condition depends on the actual implementation of d and q. For example, if a user asks for documents that do not contain "information retrieval" then the system should not return documents talking about "information retrieval", because in that case $d \wedge q$ will be false F. More formally,

$$\begin{gathered} [M_d \cap M_q \neq \phi] \Leftrightarrow \left[\exists s_i \in S_d, \exists s_j \in S_q, M_{s_i} \cap M_{s_j} \neq \phi\right] \\ \text{and} \\ \left[M_{s_i} \cap M_{s_j} \neq \phi\right] \Leftrightarrow \left[\left(A^+_{s_i} \cap A^-_{s_j} = \phi\right) \wedge \left(A^-_{s_i} \cap A^+_{s_j} = \phi\right)\right] \end{gathered} \tag{5.1}$$

  In the rest of this study,

$$\begin{gathered} [C.1 = T] \Leftrightarrow [M_d \cap M_q \neq \phi] \\ [C.1 = F] \Leftrightarrow [M_d \cap M_q = \phi] \end{gathered} \tag{5.2}$$

- **C.2**: $(\underline{m_q \subseteq m_d})$ on the one hand, this condition corresponds to the ordering relation $\leq$ defined on B_C (Theorem 4.1). On the other hand, the verification of this condition depends on the number of clauses $|S_d|$ and $|S_q|$ in d and q, respectively. According to that we have the following cases, where we suppose that $C.1 = T$ (5.2) as a pre-condition:
  - *case 1* ($|S_d| = |S_q| = 1$): suppose that $S_d = \{s_d\}$ and $S_q = \{s_q\}$ then

$$[\models d \rightarrow q] \Leftrightarrow \left[m_{s_q} \subseteq m_{s_d}\right] \tag{5.3}$$

    see Theorem 4.2.

- *case 2* ($|S_d| = 1$ and $|S_q| > 1$): suppose that $S_d = \{s_d\}$ then

$$[\models d \rightarrow q] \Leftrightarrow [\exists s_i \in S_q, m_{s_i} \subseteq m_{s_d}] \tag{5.4}$$

because for any logical sentences s_1, s_2, s_3, we have:

$$[s_1 \rightarrow (s_2 \vee s_3)] \Leftrightarrow [(s_1 \rightarrow s_2) \vee (s_1 \rightarrow s_3)]$$

- *case 3* ($|S_d| > 1$ and $|S_q| = 1$): suppose that $S_q = \{s_q\}$ then

$$[\models d \rightarrow q] \Leftrightarrow \left[\forall s_i \in S_d, m_{s_q} \subseteq m_{s_i}\right] \tag{5.5}$$

because for any logical sentences s_1, s_2, s_3, we have:

$$[(s_1 \vee s_2) \rightarrow s_3] \Leftrightarrow [(s_1 \rightarrow s_3) \wedge (s_2 \rightarrow s_3)]$$

- *case 4* ($|S_d| > 1$ and $|S_q| > 1$):

$$[\models d \rightarrow q] \Leftrightarrow \left[\forall s_i \in S_d, \exists s_j \in S_q, m_{s_j} \subseteq m_{s_i}\right] \tag{5.6}$$

because for any logical sentences s_1, s_2, s_3, s_4, we have:

$$[(s_1 \vee s_2) \rightarrow (s_3 \vee s_4)] \Leftrightarrow \begin{bmatrix} [(s_1 \rightarrow s_3) \vee (s_1 \rightarrow s_4)] \\ \wedge \\ [(s_2 \rightarrow s_3) \vee (s_2 \rightarrow s_4)] \end{bmatrix}$$

5.3 Uncertainty

IR is an uncertain process [5]. Therefore, it is mandatory to define a measure for quantifying the implication $d \rightarrow q$, written $P(d \rightarrow q)$. It is rarely the case where d directly implies q, so we need a measure to estimate the degree to which d implies q, and then ranking documents according to the decreasing value of this measure.

According to Definitions 5.1 and 5.2, documents and queries are sets of clauses, and then they correspond to sets of nodes in B_C. Moreover, B_C is a Boolean algebra (Theorem 4.1). Knuth [13] defines the z function (2.3) on distributive lattices. The $z(x, y)$ function measures the degree to which x includes or implies y for any two distinct elements x and y of a distributive lattice. Therefore, it is possible to replace the uncertainty function P by the degree of implication function z. First we should define the z function on our two lattices B_M, and B_C.

– The lattice B_M: assume s_1 and s_2 are two logical sentences, $M_{s_1} \models s_1$ is the set of models of s_1, and $M_{s_2} \models s_2$ is the set of models of s_2,

$$z_M(M_{s_1}, M_{s_2}) = \begin{cases} 1 & \text{if} \quad M_{s_2} \subseteq M_{s_1} \\ 0 & \text{if} \quad C.1 = F \\ z_M & \text{otherwise, where} \quad 0 < z_M < 1 \end{cases} \tag{5.7}$$

$C.1 = F$ (5.2) means $M_{s_1} \cap M_{s_2} = \phi$.

– The lattice B_C: assume s_1 and s_2 are two clauses in C_A, where m_{s_1} is the full clause model of s_1 and m_{s_2} is the full clause model of s_2,

$$z_C(m_{s_1}, m_{s_2}) = \begin{cases} 1 & \text{if} \quad m_{s_2} \subseteq m_{s_1} \\ 0 & \text{if} \quad C.1 = F \\ z_C & \text{otherwise, where} \quad 0 < z_C < 1 \end{cases} \tag{5.8}$$

$C.1 = F$ (5.2) means $(A^+_{s_1} \cap A^-_{s_2} \neq \phi) \vee (A^-_{s_1} \cap A^+_{s_2} \neq \phi)$. The condition $C.1$ can not be directly verified using the lattice B_C. It needs external notions (A^+ and A^-) to be verified.

Let us now come back to our initial uncertain implication $P(d \rightarrow q)$. d and q are, in general, logical sentences. By rewriting d and q in their DNF form, each of them corresponds to one or several clauses in C_A. Therefore, it is possible to rewrite P in terms of z_M (5.7) or z_C (5.8), as follow:

– $P(d \rightarrow q)$ corresponds to $z_M(M_q, M_d)$ because $z_M(M_q, M_d) = 1$ when $M_d \subseteq M_q$ which is equivalent to $\models d \rightarrow q$.
– if $(M_d \cap M_q \neq \phi)$ then $P(d \rightarrow q)$ corresponds to $z_C(m_d, m_q)$ because $z_C(m_d, m_q) = 1$ when $d \rightarrow q$ is true (Theorem 4.2), where $d, q \in C_A$, m_d is the full clause model of d, m_q is the full clause model, $M_d \models d$ is the set of models of d, and $M_q \models q$ is the set of models of q. This is true when d and q correspond to two distinct clauses in B_C, but it could be easily generalized in the other cases as we will see in the next section.

5.4 The Relevance Status Value $RSV(d, q)$

In this section, we will depend on the full clause models Boolean algebra B_C (Theorem 4.1), and we will study the different cases of d and q. For simplifying the notation, we will refer to z_C by z.

Nie [2] differentiates between the two implications *Exhaustivity* $d \rightarrow q$ and *Specificity* $q \rightarrow d$. He supposes that the matching score $RSV(d, q)$ is written as follow:

$$RSV(d, q) = F\left[P(d \rightarrow q), P(q \rightarrow d)\right] \tag{5.9}$$

We take this general form of matching score (5.9), and we build our discussion on it. According to the case where d and q correspond to only one clause ($|S_d| = |S_q| = 1$) or a set of clauses ($|S_d| > 1$ and $|S_q| > 1$) in B_C, we have: in this discussion, we suppose that $C.1 = T$,

1. *case 1* ($|S_d| = |S_q| = 1$): suppose that $S_d = \{s_d\}$ and $S_q = \{s_q\}$ then
   – $P(d \rightarrow q) = z(m_{s_d}, m_{s_q})$
   – $P(q \rightarrow d) = z(m_{s_q}, m_{s_d})$

$$\begin{aligned} RSV(d, q) &= F\left[P(d \rightarrow q), P(q \rightarrow d)\right] \\ &= F\left[z(m_{s_d}, m_{s_q}), z(m_{s_q}, m_{s_d})\right] \end{aligned} \tag{5.10}$$

2. *case 2* ($|S_d| = 1$ and $|S_q| > 1$): suppose that $S_d = \{s_d\}$ then

- $P(d \to q) = \underset{s_i \in S_q}{G} (z(m_{s_d}, m_{s_i}))$, where $G : \mathbb{R}^n \to \mathbb{R}$ and G respects the condition (5.4). For example, G could the normal sum $+$ or the max function.
- $P(q \to d) = \underset{s_i \in S_q}{G'} (z(m_{s_i}, m_{s_d}))$, where $G' : \mathbb{R}^n \to \mathbb{R}$ and G' respects the condition (5.5). For example, G' could the normal product $\times$ or the min function.

$$\begin{aligned} RSV(d,q) &= F\left[P(d \to q), P(q \to d)\right] \\ &= F\left[\underset{s_i \in S_q}{G} (z(m_{s_d}, m_{s_i})), \underset{s_i \in S_q}{G'} (z(m_{s_i}, m_{s_d}))\right] \end{aligned} \tag{5.11}$$

3. *case 3* ($|S_d| > 1$ and $|S_q| = 1$): suppose that $S_q = \{s_q\}$ then
   - $P(d \to q) = \underset{s_i \in S_d}{G'} \left(z(m_{s_i}, m_{s_q})\right)$
   - $P(q \to d) = \underset{s_i \in S_d}{G} \left(z(m_{s_q}, m_{s_i})\right)$

$$\begin{aligned} RSV(d,q) &= F\left[P(d \to q), P(q \to d)\right] \\ &= F\left[\underset{s_i \in S_d}{G'} \left(z(m_{s_i}, m_{s_q})\right), \underset{s_i \in S_d}{G} \left(z(m_{s_q}, m_{s_i})\right)\right] \end{aligned} \tag{5.12}$$

4. *case 4* ($|S_d| > 1$ and $|S_q| > 1$):
   - we define $G''(S_d; S_q) = \underset{s_i \in S_d}{G'} \circ \underset{s_j \in S_q}{G}$ where

$$G''(S_d; S_q) = \underset{s_i \in S_d}{G'} \left(\underset{s_j \in S_q}{G} \left(z(m_{s_i}, m_{s_j})\right) \right)$$

   - $P(d \to q) = G''(S_d; S_q)$
   - $P(q \to d) = G''(S_q; S_d)$

$$RSV(d,q) = F\left[P(d \to q), P(q \to d)\right] = F\left[G''(S_d; S_q), G''(S_q; S_d)\right] \tag{5.13}$$

Equation 5.13 is the most general form of the matching score $RSV(d,q)$ between a document d and a query q.

6 Discussion

After presenting our general IR model (Equation 5.13), it is now possible to discuss some results.

6.1 Result 1

If d and q are two logical sentences then they correspond to two distinct sets of models $M_d, M_q \in 2^{\Delta'_A}$, or equivalently two distinct nodes in B_M. We also know that B_M is a Boolean lattice, so the $z_M(x,y)$ function is the probability $P_M(x|y)$.

We already said that $P(d \rightarrow q)$ corresponds to $z_M(M_q, M_d)$ (Equation 5.7), so $P(d \rightarrow q)$ corresponds to $P_M(M_q|M_d)$. However, we know that each node $M \in 2^{\Delta'_A}$ in the lattice B_M is a set of models of a set of equivalent logical sentences. By this way, M_q is a set of models of a set of logical sentences equivalent to q. We choose q as a representative to this equivalent class. We do the same thing for d. Therefore,

$$P(d \rightarrow q) = P(q|d) \tag{6.1}$$

Equation (6.1) justifies the definition of $P(d \rightarrow q)$ that is presented by Rijsbergen [1]. To our knowledge, this is the first study that present a formal justification of the Rijsbergen's intuition.

6.2 Result 2

Assume that d and q correspond to two distinct full clause models $m_d, m_q \in \Psi_A$, or equivalently two distinct nodes m_d and m_q in B_C. The results that we will obtain could be easily generalized to the cases where d and q are two sets of nodes.

Exhaustivity $P(d \rightarrow q)$ corresponds to $z(m_d, m_q)$ in B_C (Equation 5.8). Using the first product rule (2.5), we obtain:

$$z(m_d, m_q) = z(m_q \wedge m_d, m_q) \tag{6.2}$$

The same for Specificity $P(q \rightarrow d)$:

$$z(m_q, m_d) = z(m_d \wedge m_q, m_d) \tag{6.3}$$

Those two results (6.2) and (6.3) correspond to our intuition presented in our previous study [11]. Depending on the Bay's Theorem rule (2.7), it is possible to deduce that:

$$z(m_d, m_q) = \frac{z(m_d, \top)}{z(m_q, \top)} \times z(m_q, m_d) \tag{6.4}$$

where $z(x, \top)$ represent prior probabilities, which can be arbitrary assigned [13]. Using the Sum rule (2.4), we obtain:

$$z(m_d, m_q) = 1 - z(\neg m_d, m_q) \tag{6.5}$$

Actually, there are many interesting properties could be deduced from playing on the rules of the z function (2.4, 2.5, 2.6, and 2.7).

6.3 Result 3

We think that (5.13) forms a general IR framework and some well-known IR models could be derived from (5.13).

Language Models (LM). Assume that each proposition $a_i \in A$ corresponds to one term in the document collection D. A document $d \in D$ is written as follow:

$$d = \bigwedge_{a_i \in A_d} a_i \tag{6.6}$$

where $A_d \neq \phi$ and a_i is the proposition that corresponds to the term t_i that occurs in d. In other words, d is a conjunction of the terms that occur in it. For any document d, we have: $A_d^- = \phi$, or in other words, the negation of terms is not modelized. The query q is represented in the same way,

$$q = \bigwedge_{a_i \in A_q} a_i \tag{6.7}$$

Hence, $d, q \in C_A$ are two clauses, and they correspond to two distinct nodes m_d and m_q in B_C, respectively. Therefore, $G''(\{d\}; \{q\}) = z(m_d, m_q)$ and also $G''(\{q\}; \{d\}) = z(m_q, m_d)$.

We choose F as the Weighted-Sum of two values, so (5.13) could be rewritten as follows:

$$RSV(d, q) = \alpha \times z(m_d, m_q) + \beta \times z(m_q, m_d)$$

Now, assume that $\alpha = 0$ and $\beta = 1$ (Specificity without Exhaustivity) then

$$RSV(d, q) = z(m_q, m_d)$$

As B_C is a Boolean lattice, it is possible to replace z by the probability P_C then

$$RSV(d, q) = P_C(m_q | m_d)$$

Now, suppose that the elements of m_q are conditionally independent, and let us define a probability distribution θ_d on the set m_d then

$$RSV(d, q) = \prod_{a_i \in m_q} P_C(a_i | \theta_d)$$

which is the general form of language models. Therefore, language models are instances of our general framework.

Probabilistic Models (PM). Probabilistic Models (PMs) depend on the Probability Ranking Principle [15], according to which: documents are ranked according to the decreasing value of the probability $P(R|d, q)$. More precisely, PMs use the notion of odds (6.8).

$$RSV(d, q) \propto \frac{P(R|d, q)}{P(NR|d, q)} \tag{6.8}$$

R means *document is relevant* whereas NR means *document is non-relevant.*

The main problem in PMs is that the relevance information is not available in advance. Therefore, it is hard to estimate the two probabilities $P(R|d, q)$ and

$P(NR|d,q)$. However, using Bay's rule and with some simplifications, the ranking formula (6.8) becomes:

$$RSV(d,q) \propto \frac{P(d|R,q)}{P(d|NR,q)} \tag{6.9}$$

To estimate the two probabilities, we should have samples of the relevant and non-relevant documents of each query.

Some studies [16, 17] claim that R and NR could be seen as two sets of relevant and non-relevant documents for a specific query, respectively. Robertson et al. [18] also assume that a set of relevance judgements for each request should be available to estimate the relevance weights. If we take this viewpoint, where R and NR are two sets of documents, and project it on the lattice B_C, we obtain:

- assume that each proposition $a_i \in A$ corresponds to one term in the document collection D.
- any document $d \in D$ has the same definition presented in LM (6.6), so it corresponds to only one node m_d in B_C.
- any query q has the same definition presented in LM (6.7), so it corresponds to only one node m_q in B_C.
- R is a disjunction of a set of documents $R = d_1 \vee \cdots \vee d_k$ where any document d_i satisfies $\models d_i \rightarrow q$. That is correct because $\models R \rightarrow q$ means that for any d_i we have $\models d_i \rightarrow q$. Therefore, R corresponds to a set of nodes N_R in B_C.

$$N_R = \{m_{d_i} | 1 \leq i \leq k, \models d_i \rightarrow q\}$$

- NR is also a disjunction of a set of documents $NR = d_1 \vee \cdots \vee d_l$ where any document d_i satisfies $\not\models d_i \wedge q$. That is correct because $\not\models NR \wedge q$ means that for any d_i we have $\not\models d_i \wedge q$. Therefore, NR corresponds to a set of nodes N_{NR} in B_C.

$$N_{NR} = \{m_{d_i} | 1 \leq i \leq l, \not\models d_i \wedge q\}$$

- The retrieval decision could be reformulated as follow: d is relevant to q *if* $\models d \rightarrow R$ and $\not\models d \rightarrow NR$ (Theorem 6.1). By taking the degree of implication z into account, we have:

$$RSV(d,q) \propto \frac{G''(\{d\};R)}{G''(\{d\};NR)}$$

$G''(\{d\};R)$ could be simplified to $\underset{r \in R}{G}(z(m_d, m_r))$, and we choose the max function to replace G (5.11). Moreover, as B_C is a Boolean algebra then $z(x,y)$ could be replaced by the probability $P_C(x|y)$. We also suppose that the elements of m_d are conditionally independent. Finally, we obtain the following ranking formula:

$$RSV(d,q) \propto \prod_{a_i \in m_d} \frac{P_C(a_i|m_R)}{P_C(a_i|m_{NR})}$$

where m_R is the node that maximises $\underset{r \in R}{G}(z(m_d, m_r))$, and m_{NR} is the node that maximises $\underset{m_{nr} \in NR}{G}(z(m_d, m_{nr}))$.

The previous formula is the general form of probabilistic models. Therefore, probabilistic models are instances of our general framework.

Theorem 6.1. *A document d is relevant to a query q if:*

$$\models (d \rightarrow R) \quad \textit{and} \quad \not\models (d \rightarrow NR)$$

Proof. Assume $R = d_1^R \vee \cdots \vee d_k^R$, where $\forall 1 \leq i \leq k, \models d_i^R \rightarrow q$. Assume $NR = d_1^{NR} \vee \cdots \vee d_l^{NR}$, where $\forall 1 \leq i \leq l, \not\models d_i^{NR} \wedge q$.

$\models d \rightarrow R$ means that $\exists 1 \leq i \leq k, \models d \rightarrow d_i^R$, which in its turn means $\models d \rightarrow q$ because $\models R \rightarrow q$. Suppose that $\models d \rightarrow NR$ then:

- $\models (d \rightarrow d_1^{NR}) \vee \cdots \vee (d \rightarrow d_1^{NR})$
- $\exists 1 \leq i \leq l, \models d \rightarrow d_i^{NR}$
- $\models \neg d \vee d_i^{NR}$
- *we know that* $\not\models d_i^{NR} \wedge q$ *then:* $\not\models \neg(\neg d \vee d_i^{NR}) \vee (d_i^{NR} \wedge q)$
- $\not\models \big((d \wedge \neg d_i^{NR}) \vee d_i^{NR}\big) \wedge \big((d \wedge \neg d_i^{NR}) \vee q\big)$
- $\not\models (d \vee q) \wedge \big((d \vee d_i^{NR}) \wedge (\neg d_i^{NR} \vee q)\big)$
- $\not\models (d \vee q) \wedge (d \wedge q)$
- $\not\models (d \wedge q)$

We should thus change our hypothesis from ($\models d \rightarrow NR$) to ($\not\models d \rightarrow NR$).

□

Unlike the previous implementations of PMs, lattices allow us to define the two sets R and NR in advance. We first define the up-set of a node m in our lattice B_C, noted $\uparrow m$:

$$\forall m \in \Psi_A, \uparrow m = \{m' | m' \in \Psi_A, m \leq m'\}$$

We define the non-relevant nodes N_{NR} in B_C:

$$N_{NR} = \{m | m \in \Psi_A, z(m, m_q) = 0\}$$

Now, it is possible to define the set of nodes N_R,

$$N_R = (\uparrow m_q) \setminus N_{NR}$$

The lattice B_C allows us to define the family of probabilistic models. Moreover, it allows us to determine the relevant and non-relevant documents in advance, which are very important to estimate the two probabilities $P(d|R, q)$ and $P(d|NR, q)$.

Vector Space Model (VSM). Assume that each proposition $a_i \in A$ corresponds to one term in the document collection D. Suppose that any document $d \in D$ has the same definition presented in LM (6.6), so d corresponds to only one node m_d in B_C. Suppose that any query q has the same definition presented in LM (6.7), so q corresponds to only one node m_q in B_C.

For any node $m \in \Psi_A$ in B_C, it is possible to build a binary vector $\overrightarrow{m}$ as follow:

$$\overrightarrow{m} = \langle w(a_1), \ldots, w(a_n) \rangle$$

where $w(a_i) = 1$ if $a_i \in m$, or $w(a_i) = 0$ otherwise. For simplicity, we will refer to $w(a_i)$ by w_i. We define the following two operations:

1. Production: $\forall m_1, m_2 \in \Psi_A$,

$$\overrightarrow{m_1} \otimes \overrightarrow{m_2} = \langle w_1^1 \times w_1^2, \ldots, w_n^1 \times w_n^2 \rangle$$

The $\otimes$ operation between the two vectors $\overrightarrow{m_1}$ and $\overrightarrow{m_2}$ corresponds to the meet operation $\wedge$ between the two nodes m_1 and m_2, where,

$$\overrightarrow{m_1 \wedge m_2} = \overrightarrow{m_1} \otimes \overrightarrow{m_2}$$

2. Addition: $\forall m_1, m_2 \in \Psi_A$,

$$\overrightarrow{m_1} \oplus \overrightarrow{m_2} = \langle w_1^1 + w_1^2 - w_1^1 \times w_1^2, \ldots, w_n^1 + w_n^2 - w_n^1 \times w_n^2 \rangle$$

The $\oplus$ operation between the two vectors $\overrightarrow{m_1}$ and $\overrightarrow{m_2}$ corresponds to the join operation $\vee$ between the two nodes m_1 and m_2, where,

$$\overrightarrow{m_1 \vee m_2} = \overrightarrow{m_1} \oplus \overrightarrow{m_2}$$

Let us define the z function as the inner-product $(\cdot)$ between two nodes, as follow:

$$z(m_1, m_2) = \begin{cases} 0 & \text{if} \quad C.1 = F \\ \frac{\overrightarrow{m_1} \cdot \overrightarrow{m_2}}{|\overrightarrow{m_2}|} & \text{otherwise} \end{cases} \tag{6.10}$$

where $|\overrightarrow{m}| = \sum_{t_i} w_i$. However, the z function (6.10) satisfies the sum rule of the z function (2.4), where:

$$\frac{(\overrightarrow{x} \oplus \overrightarrow{y}) \cdot \overrightarrow{t}}{|\overrightarrow{t}|} = \frac{\overrightarrow{x} \cdot \overrightarrow{t}}{|\overrightarrow{t}|} + \frac{\overrightarrow{y} \cdot \overrightarrow{t}}{|\overrightarrow{t}|} - \frac{(\overrightarrow{x} \otimes \overrightarrow{y}) \cdot \overrightarrow{t}}{|\overrightarrow{t}|} \tag{6.11}$$

We choose the classical product $(\times)$ instead of F, so (5.13) could be rewritten as follows:

$$RSV(d, q) = z(m_d, m_q) \times z(m_q, m_d)$$

According to (6.2) and (6.3), it is possible to rewrite the previous equation as follow:

$$RSV(d, q) = z(m_d \wedge m_q, m_q) \times z(m_d \wedge m_q, m_d)$$

which is compatible with our previous publication [11].

7 Conclusion

We present in this study a new theoretical framework for representing documents, queries, and the retrieval decision including a ranking mechanism. We use the Propositional Logic for representing documents and queries, and then we claim that the retrieval decision corresponds to the material implication between a document and a query. After that, we position documents and queries on a lattice, more precisely on a Boolean algebra. We then exploit the degree of implication metric z, defined on the lattice, for representing the ranking mechanism.

This study, on the one hand, presents a new vision of logic-based IR models through exploiting the implicit link between lattices and Propositional Logic. On the other hand, it presents a general IR framework capable of representing the classical IR models, like Language Models, Probabilistic Models, and Vector Space Models.

Our model also provides a theoretical proof for the definition of $P(d \rightarrow q)$ that is presented by Rijsbergen [1]. In addition, it provides a theoretical proof for the choices that are made in [11].

The most important point in this study, in our point of view, is the simplicity and flexibility of the framework that it provides. We discussed a few capabilities of our model, but there still exists so many potential capabilities waiting to be discovered, especially through working on the rules of the z function (2.4, 2.5, 2.6, and 2.7), and through exploiting the partial order relation defined on the Boolean algebra B_C.

References

1. van Rijsbergen, C.J.: A non-classical logic for information retrieval. Comput. J. **29**(6) (1986) 481–485
2. Nie, J.: An outline of a general model for information retrieval systems. SIGIR '88, New York, NY, USA, ACM (1988) 495–506
3. Nie, J.: An information retrieval model based on modal logic. Information Processing & Management **25**(5) (1989) 477 – 491
4. Nie, J.Y.: Towards a probabilistic modal logic for semantic-based information retrieval. In: Proceedings of the 15th annual international ACM SIGIR conference on Research and development in information retrieval. SIGIR '92, New York, NY, USA, ACM (1992) 140–151
5. Chiaramella, Y., Chevallet, J.P.: About retrieval models and logic. Comput. J. **35** (June 1992) 233–242
6. Meghini, C., Sebastiani, F., Straccia, U., Thanos, C.: A model of information retrieval based on a terminological logic. In: Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval. SIGIR '93, New York, NY, USA, ACM (1993) 298–307
7. Sebastiani, F.: A probabilistic terminological logic for modelling information retrieval. In: Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval. SIGIR '94, New York, NY, USA, Springer-Verlag New York, Inc. (1994) 122–130

8. Chevallet, J.P., Chiaramella, Y.: Experiences in information retrieval modelling using structured formalisms and modal logic. In Crestani, F., Lalmas, M., Rijsbergen, C., eds.: Information Retrieval: Uncertainty and Logics. Volume 4 of The Kluwer International Series on Information Retrieval. Springer US (1998) 39–72
9. Crestani, F., Lalmas, M.: Logic and uncertainty in information retrieval. In: Proceedings of the Third European Summer-School on Lectures on Information Retrieval-Revised Lectures. ESSIR '00, London, UK, UK, Springer-Verlag (2001) 179–206
10. Losada, D.E., Barreiro, A.: A logical model for information retrieval based on propositional logic and belief revision. The Computer Journal **44** (2001) 410–424
11. Abdulahhad, K., Chevallet, J.P., Berrut, C.: The effective relevance link between a document and a query. In Liddle, S., Schewe, K.D., Tjoa, A., Zhou, X., eds.: Database and Expert Systems Applications. Volume 7446 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2012) 206–218
12. Knuth, K.H.: Deriving laws from ordering relations. In: In press: Bayesian Inference and Maximum Entropy Methods in Science and Engineering, Jackson Hole WY. (2003) 204–235
13. Knuth, K.H.: Lattice duality: The origin of probability and entropy. Neurocomput. **67** (August 2005) 245–274
14. Dominich, S.: The Modern Algebra of Information Retrieval. 1 edn. Springer Publishing Company, Incorporated (2008)
15. Robertson, S.E.: The Probability Ranking Principle in IR. Journal of Documentation **33**(4) (1977) 294–304
16. Hiemstra, D., Vries, A.P.D.: Relating the new language models of information retrieval to the traditional retrieval models. Technical report (2000)
17. Lavrenko, V., Croft, W.B.: Relevance based language models. In: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval. SIGIR '01, New York, NY, USA, ACM (2001) 120–127
18. Robertson, S.E., Jones, K.S.: Relevance weighting of search terms. J. Am. Soc. Inf. Sci. **27**(3) (1976) 129–146