Information Retrieval (IR) Modeling by Logic and Lattice. Application to Conceptual IR

Karam Abdulahhad

UNIVERSITÉ DE GRENOBLE

THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ DE GRENOBLE

Spécialité : **Informatique**

Arrêté ministériel : 7 août 2006

Présentée par

Karam ABDULAHHAD

Thèse dirigée par **Catherine BERRUT** et
codirigée par **Jean-Pierre CHEVALLET**

préparée au sein du **Laboratoire LIG**
dans **l'École Doctorale MSTII**

Information Retrieval (IR) Modeling by Logic and Lattice

Application to Conceptual IR

Thèse soutenue publiquement le **« 05 mai 2014 »**,
devant le jury composé de :

Prof. Christine VERDIER
Université Joseph Fourier (Président)
Prof. Fabio CRESTANI
University of Lugano (Rapporteur)
Prof. Jian-Yun NIE
Université de Montréal (Rapporteur)
Dr. Vincent CLAVEAU
CNRS-IRISA (Membre)
Prof. Catherine BERRUT
Université Joseph Fourier (Membre)
Dr. Jean-Pierre CHEVALLET
Université Pierre Mendès France (Membre)

Abstract

This thesis is situated in the context of logic-based Information Retrieval (IR) models. The work presented in this thesis is mainly motivated by the inadequate term-independence assumption, which is well-accepted in IR although terms are normally related, and also by the inferential nature of the relevance judgment process. Since formal logics are well-adapted for knowledge representation, and then for representing relations between terms, and since formal logics are also powerful systems for inference, logic-based IR thus forms a candidate piste of work for building effective IR systems. However, a study of current logic-based IR models shows that these models generally have some shortcomings. First, logic-based IR models normally propose complex, and hard to obtain, representations for documents and queries. Second, the retrieval decision $d \rightarrow q$, which represents the matching between a document d and a query q, could be difficult to verify or check. Finally, the uncertainty measure $U(d \rightarrow q)$ is either ad-hoc or hard to implement.

In this thesis, we propose a new logic-based IR model to overcome most of the previous limits. We use Propositional Logic ($\mathcal{PL}$) as an underlying logical framework. We represent documents and queries as logical sentences written in Disjunctive Normal Form. We also argue that the retrieval decision $d \rightarrow q$ could be replaced by the validity of material implication $\models d \supset q$. We then exploit the potential relation between $\mathcal{PL}$ and lattice theory to check if $d \supset q$ is valid or not. We first propose an intermediate representation of logical sentences, where they become nodes in a lattice having a partial order relation that is equivalent to the validity of material implication. Accordingly, we transform the checking of $\models d \supset q$, which is a computationally intensive task, to a series of simple set-inclusion checking. In order to measure the uncertainty of the retrieval decision $U(d \rightarrow q)$, we use the degree of inclusion function Z that is capable of quantifying partial order relations defined on lattices. Finally, our model is capable of working efficiently on any logical sentence without any restrictions, and is applicable to large-scale data. Our model also has some theoretical conclusions, including, formalizing and showing the adequacy of van Rijsbergen assumption about estimating the logical uncertainty $U(d \rightarrow q)$ through the conditional probability $P(q|d)$, redefining the two notions Exhaustivity & Specificity, and the possibility of reproducing most classical IR models as instances of our model.

We build three operational instances of our model. An instance to study the importance of Exhaustivity and Specificity, and two others to show the inadequacy of the term-independence assumption. Our experimental results show worthy gain in performance when integrating Exhaustivity and Specificity into one concrete IR model. However, the results of using semantic relations between terms were not sufficient to draw clear conclusions. On the contrary, experiments on exploiting structural relations between terms were promising. The work presented in this thesis can be developed either by doing more experiments, especially about using relations, or by more in-depth theoretical study, especially about the properties of the Z function.

Résumé

Cette thèse se situe dans le contexte des modèles logique de Recherche d'Information (RI). Le travail présenté dans la thèse est principalement motivé par l'inexactitude de l'hypothèse sur l'indépendance de termes. En effet, cette hypothèse communément acceptée en RI stipule que les termes d'indexation sont indépendant les un des autres. Cette hypothèse est fausse en pratique mais permet toit de même aux systèmes de RI de donner de bon résultats. La proposition contenue dans cette thèse met également l'emphase sur la nature déductive du processus de jugement de pertinence. Les logiques formelles sont bien adaptées pour la représentation des connaissances. Elles permettent ainsi de représenter les relations entre les termes. Les logiques formelles sont également des systèmes d'inférence, ainsi la RI à base de logique constitue une piste de travail pour construire des systèmes efficaces de RI. Cependant, en étudiant les modèles actuels de RI basés sur la logique, nous montrons que ces modèles ont généralement des lacunes. Premièrement, les modèles de RI logiques proposent normalement des représentations complexes de document et des requête et difficile à obtenir automatiquement. Deuxièmement, la décision de pertinence $d \rightarrow q$, qui représente la correspondance entre un document d et une requête q, pourrait être difficile à vérifier. Enfin, la mesure de l'incertitude $U(d \rightarrow q)$ est soit ad-hoc ou difficile à mettre en oeuvre.

Dans cette thèse, nous proposons un nouveau modèle de RI logique afin de surmonter la plupart des limites mentionnées ci-dessus. Nous utilisons la logique propositionnelle ($\mathcal{PL}$). Nous représentons les documents et les requêtes comme des phrases logiques écrites en Forme Normale Disjonctive. Nous argumentons également que la décision de pertinence $d \rightarrow q$ pourrait être remplacée par la validité de l'implication matérielle $\models d \supset q$. Pour vérifier si $d \supset q$ est valide ou non, nous exploitons la relation potentielle entre $\mathcal{PL}$ et la théorie des treillis. Nous proposons d'abord une représentation intermédiaire des phrases logiques, où elles deviennent des noeuds dans un treillis ayant une relation d'ordre partiel équivalent à la validité de l'implication matérielle. En conséquence, nous transformons la vérification de $\models d \supset q$, ce qui est un calcul intensif, en une série de vérifications simples d'inclusion d'ensembles. Afin de mesurer l'incertitude de la décision de pertinence $U(d \rightarrow q)$, nous utilisons la fonction du degré d'inclusion Z, qui est capable de quantifier les relations d'ordre partielles définies sur des treillis. Enfin, notre modèle est capable de travailler efficacement sur toutes les phrases logiques sans aucune restriction, et est applicable aux données à grande échelle. Notre modèle apporte également quelques conclusions théoriques comme: la formalisation de l'hypothèse de van Rijsbergen sur l'estimation de l'incertitude logique $U(d \rightarrow q)$ en utilisant la probabilité conditionnelle $P(q|d)$, la redéfinition des deux notions Exhaustivity & Specificity, et finalement ce modèle a également la possibilité de reproduire les modèles les plus classiques de RI.

De manière pratique, nous construisons trois instances opérationnelles de notre modèle. Une instance pour étudier l'importance de Exhaustivity et Specificity, et deux autres pour montrer l'insuffisance de l'hypothèse sur l'indépendance des termes. Nos résultats expérimentaux montrent un gain de performance lors de l'intégration Exhaustivity et Specificity. Cependant,

les résultats de l'utilisation de relations sémantiques entre les termes ne sont pas suffisants pour tirer des conclusions claires. Le travail présenté dans cette thèse doit être poursuivit par plus d'expérimentations, en particulier sur l'utilisation de relations, et par des études théoriques en profondeur, en particulier sur les propriétés de la fonction Z.

Contents

List of Figures

List of Tables

Part I

INTRODUCTION

Chapter 1

Introduction

1.1 General Context

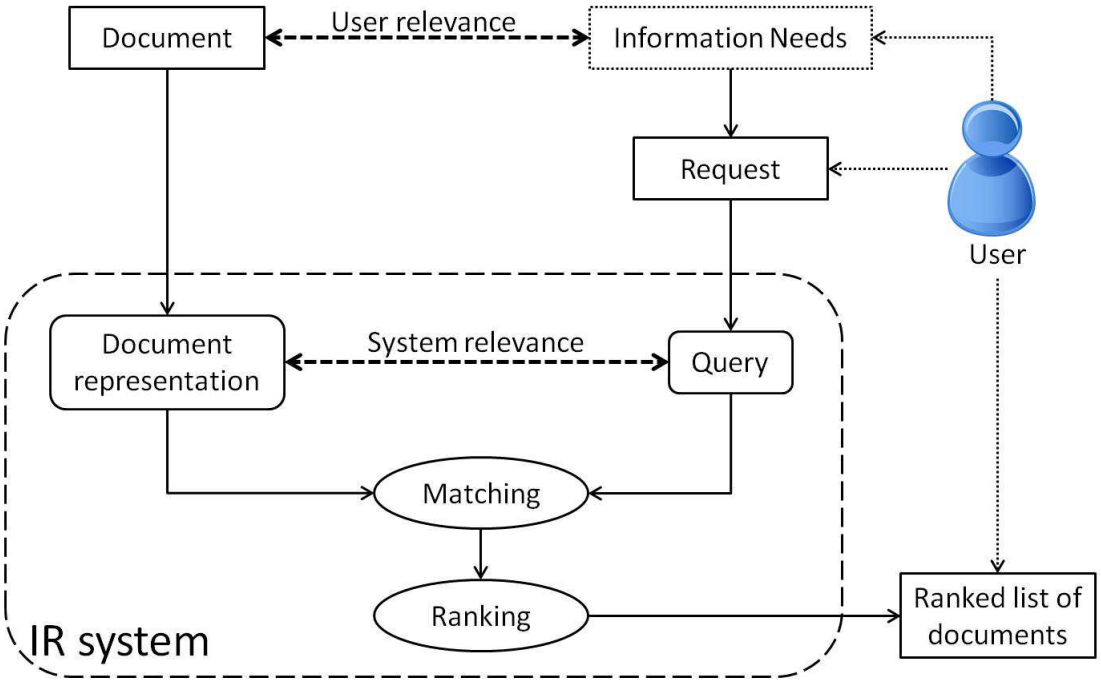
Information Retrieval (IR) systems play an essential role in the current information age. More available and freely accessible information means greater need, or even indispensable need, to automate the process of information representation and retrieval. Nowadays, in most cases, information is organized into documents, e.g. web pages, books, etc. Users have information needs or information gaps that need to be filled. Therefore, users search the document collection or corpus in order to find the documents that fulfill, from users' point of view, their information needs.

Figure 1.1 shows a general view of IR systems, where a *user* has *information needs* and he/she normally translates these needs to a *request* in his/her natural language. The role of any IR system is then to build a *document representation* and a *query* which are the machine-adapted version of the document and the request, respectively. This operation is normally called indexing. After that, the IR system compares the query with the document representation in order to establish the *matching* between them, or to decide if the document is relevant to the query. Finally, the IR system depends on a *ranking* mechanism to distinguish between the more and the less relevant documents to a query. In (Figure 1.1), we separate the two processes of matching and ranking, but some IR models combine these two processes. We can also see that the users' point of view about relevance is different from the system's point of view, where the former is between the original information needs and the original document, whereas the latter is between the query and the document representation.

In this thesis, we do not address the user-system interaction issue, it is actually beyond the scope of this work. We rather consider the part that does not require a direct user involvement, namely the dashed box (IR system) in (Figure 1.1). To build an automated IR system, we should first build a machine-adapted document representation. We should also build queries, which are the machine-adapted representation of user's requests. Finally, and in order to make the IR system being capable of retrieving relevant documents, we should define the system's point of view about relevance.

Beyond IR systems lies IR models, which formally determine the way in which information must be represented and retrieved. In general, any IR model must clearly define four main components or notions: a document representation, a query, a relevance or a retrieval decision

Figure 1.1: A general view of IR systems



from the system's point of view, and a ranking mechanism for finally the system being able to build a document-preference.

1.1.1 Logic-Based IR Models

Some researchers propose to use formal logics to represent the whole IR process. The actual starting point was the paper of van Rijsbergen [van Rijsbergen, 1986]. He argues that if a document d is a set of logical sentences in a specific logic and a query q is a logical sentence in the same logic, then the retrieval decision or the relevance between d and q can be formalized through a logical implication $d \rightarrow q$, where d should be retrieved as an answer to q *iff* d logically implies q. However, using the logical implication $d \rightarrow q$ for representing the retrieval decision is quite limited, because $d \rightarrow q$ is a binary decision, i.e. either d implies q or not, or in other words, either d is relevant to q or not. A binary decision cannot model the fact that IR is intrinsically an uncertain process[1] , where:

- the query q is an imperfect representation of user needs;
- the document d is also an imperfect representation of the content of documents because the indexing process is always imperfect;
- relevance judgment depends on external factors, like user background knowledge, or in other words, relevance judgment is a user-dependent decision, i.e. the decision that d is relevant or not to q depends on the user who asks the query q.

[1] Actually, each transition or operation in (Figure 1.1) is a potential source of uncertainty.

We thus need a more flexible notion of implication between d and q for reflecting this *uncertainty* or imperfection. We need to estimate the degree of implication or the uncertainty of implication, denoted $U(d \rightarrow q)$.

Using the uncertain implication $U(d \rightarrow q)$ for representing the retrieval decision means that, in logic-based IR models, there are four components need to be clearly defined: the document d, the query q, the logical implication $d \rightarrow q$, and the uncertainty function U. In general, the formal definitions of these four components are based on the chosen logic as an underling mathematical framework. In IR field, there is a variety of formal logics used as mathematical frameworks, e.g. modal propositional logic, first-order logic, description logic, etc.

The work presented in this thesis lies in the range of logic-based IR models. These models are based on a logical framework to represent documents, queries, and to express the relevance from the system's point of view.

Thesis' context: *Logic-based IR models.*

1.2 Motivations

There are two main motivations behind the work in this thesis. The first motivation is related to the inadequacy of the well-accepted *term-independence* assumption. The second motivation is related to the *inferential* nature of the retrieval decision.

1.2.1 The Term-Independence Assumption

Most IR models assume that terms[1] are independent. However, this is a superficial and inadequate hypothesis because terms are normally connected to each others via some relations. For example, assume a document d contains the term '*fiddle*' and a query q contains the term '*violin*', without any knowledge source, d and q are not related because '*fiddle*' and '*violin*' are independent, but by knowing that '*fiddle*' and '*violin*' are synonymous then d and q are likely related.

There are several approaches in IR literature trying to overcome such a type of term-mismatch problems. More precisely, they propose to establish some statistical connections between terms, e.g. co-occurrence [van Rijsbergen, 1977], and term-relatedness [Grefenstette, 1992]. This latter uses syntactic approach to define term association, and it is less noisy than the co-occurrence approach. However, these approaches are often not effective, because relations like synonymy, hyponym/hypernym, etc., can hardly be identified statistically [Nie & Brisebois, 1996]. For example, it is very rare to use two synonymous terms in the same chunk of text. In addition, using statistical techniques introduces some noise, because two statistically related terms are not forcibly truly related terms. Hence, it is vital to exploit manually-built and humanly-validated relations, which are normally defined between elements, called concepts[2], and organized within knowledge resources.

Motivation 1: *The first motivation is the inadequacy of the term-independence assumption, which leads to the term-mismatch problem.*

[1] We mean by "*term*" an indexing term.

[2] Chapter 2 reviews various concept definitions.

1.2.2 Relevance is a Process of Inference

The retrieval process has an *inferential* nature, because a document d indexed by a term t_1 is not directly relevant, from the system's point of view, to a query q indexed by another term t_2. However, if we know that t_1 and t_2 are synonymous, then based on d and this knowledge, q can be inferred. The inferential nature becomes clearer if we consider more complex relations like hyponym/hypernym. For example, assume a document d about '*dogs*', and a user asks for information about '*animals*' in general. By using traditional document and query comparison, the previous document does not fulfill user's information needs, but by knowing that '*a dog* ***is an*** *animal*' then it is possible to infer that a document about '*dogs*' could fulfill user's information needs about '*animals*'. Therefore, the simple term-based intersection between a document and a query is clearly insufficient mechanism for building effective IR models.

> **Motivation 2:** *The second motivation is the inferential nature of the retrieval process, where the classical document-query comparison paradigm is insufficient.*

1.2.3 Interesting Aspects in Formal Logics

Formal logics are well adapted for modeling knowledge and inference [Baader *et al.*, 2003]. In general, a formal logic $\mathcal{L}$ is a formal system consisting of a set of axioms and a set of inference rules. The inference mechanism is denoted by $\vdash_{\mathcal{L}}$. We say that a logical sentence s is *provable* based on a set of logical sentences Γ, denoted $\Gamma \vdash_{\mathcal{L}} s$, *iff* s can be obtained by applying the inference rules of $\mathcal{L}$ to the axioms of $\mathcal{L}$ and the set of sentences Γ. Furthermore, $\vdash_{\mathcal{L}} s$ means that s can be obtained by applying the inference rules of $\mathcal{L}$ to only the axioms of $\mathcal{L}$. For example, assume $\Gamma = \{s_1, s_1 \supset s_2\}$ where $\supset$ is the material implication, then in classical logics s_2 is provable based on Γ, denoted $\{s_1, s_1 \supset s_2\} \vdash s_2$ (Modus-Ponens).

Hence, in the IR field, assume that d and q are represented in a way compatible with $\mathcal{L}$, then the retrieval decision could be $d \vdash_{\mathcal{L}} q$, which means, by applying the inference rules of $\mathcal{L}$ to d and the set of axioms of $\mathcal{L}$, we infer q. If the knowledge Γ is also expressed in a way compatible with $\mathcal{L}$, then the retrieval decision becomes $\Gamma \cup \{d\} \vdash_{\mathcal{L}} q$, which means, by applying the inference rules of $\mathcal{L}$ to d, the knowledge Γ, and the set of axioms of $\mathcal{L}$, we infer q. In the former representation $d \vdash_{\mathcal{L}} q$, the inference of q is only based on d and $\mathcal{L}$, whereas in the latter $\Gamma \cup \{d\} \vdash_{\mathcal{L}} q$, the inference is also based on Γ, which could be a representation of particular knowledge e.g. some relations between terms.

Now, in classical logics, if we consider the *formal interpretation* of $\mathcal{L}$, then instead of talking about inference (provability), denoted $\vdash_{\mathcal{L}}$, we talk about logical consequence (satisfiability), denoted $\models_{\mathcal{L}}$. The inference process is thus equivalent to the validity of material implication, where d is relevant to q *iff* d logically implies q [Chiaramella & Chevallet, 1992; Crestani *et al.*, 1998; Lalmas, 1998]. For more information about formal logics, please refer to (Appendix C–P.185), where we explain the two operators $\vdash$ and $\models$ on Propositional Logic ($\mathcal{PL}$).

To sum up, formal logics are powerful tools for: knowledge representation, knowledge integration into IR process, and reproducing the inferential character of the retrieval decision.

1.2.4 A Candidate Work Track

On the one hand, exploiting well-defined relations between terms is supposed to be useful from an IR point of view (*Motivation 1*). Linguistic and semantic relations are normally a part of a knowledge resource like ontology, knowledge-base, or thesaurus. Thus, the formal integration of these resources into the IR process is supposed to be an effective way to improve the retrieval performance of IR models. On the other hand, the retrieval process has an inferential nature (*Motivation 2*).

Therefore, logic-based IR models are supposed to be a useful mathematical tool to build *more accurate*[1] IR models. Actually, the interesting aspect of using formal logics in IR is two-fold:

- Formal logics are well adapted for knowledge representation [Baader *et al.*, 2003; Barwise, 1989; Barwise & Perry, 1983], and then for building IR models being capable of formally integrating knowledge resources into the retrieval process [Meghini *et al.*, 1993; Nie & Brisebois, 1996].
- Formal logics are powerful tools for simulating and modeling the inferential nature of the retrieval process.

Although formal logics are powerful and important tools for building more accurate IR models, logic-based IR models were abandoned since the late nineties. The main reason of this abandonment is the numerous obstacles facing transforming theoretical logic-based IR models to operational models applicable to large-scale data.

> **Our work track:** *We choose to work in logic-based IR models due to the potentials of formal logics in inference, and in knowledge representation and integration.*

1.3 Problems to be Solved

Integrating, formally or informally, knowledge resources into IR process means we can use a more informative type of terms, namely concepts, and it also means the possibility to exploit the semantic relations between concepts. Even though we do not claim that this approach is better than classical IR approaches, we believe that the more explicit and validated knowledge is used, the more effective IR systems should we obtain. More precisely, integrating knowledge resources into the IR process represents a possible solution of a range of problems, e.g. term-mismatch [Crestani, 2000], multilingualism/multi-modality [Chevallet *et al.*, 2007], etc. However, this integration is also a source of another range of problems, e.g. text-to-concepts mapping tools are a possible source of noise [Maisonnasse *et al.*, 2009], incompleteness of knowledge resources [Bodenreider *et al.*, 1998, 2001], etc.

Actually, the main focus of this thesis is not to deal with the problems that could occur when using concepts and knowledge resources, even if we have some publications in this context [Abdulahhad *et al.*, 2011a,c, 2012b, 2013b] (see Appendix A–P.165). The main focus of this

[1] More accurate IR model means a model deciding relevance in a closer way to the human relevance judgment.

thesis is to study the shortcomings of current logic-based IR models, and to propose a logic-based IR model being capable of overcoming some of these problems and shortcomings.

As we said, to define a logic-based IR model, we need to clearly define four main components or notions: a document d, a query q, a retrieval decision or implication $d \rightarrow q$, and an uncertainty measure $U(d \rightarrow q)$. Logic-based IR models, which have been proposed in IR literature, have several limitations. We explore in the following the limitations of these models for each of the previous components.

> **Goal:** *The main purpose of this thesis is to propose a logic-based IR model that is operational and applicable to large-scale data.*

1.3.1 Document and Query Representation

Depending on the logic used to build an IR model, documents and queries are represented in various ways. Document representations[1] range, depending on the expressive power of logic, from acceptably easy to obtain to very hard to obtain. We mean by *easy to obtain* that the process of obtaining document representation is automatic and applicable to large-scale data.

Propositional Logic ($\mathcal{PL}$) based models have a fairly easy to obtain document representation, namely representing documents as a conjunction of terms. More precisely, logical sentences that represent documents are easy to obtain, but under some *restrictions*. For example, most $\mathcal{PL}$ based models suppose that the terms within a document are atomic propositions, and the only allowed logical connective between terms is the conjunction ($\wedge$). Some models theoretically deal with disjunction ($\vee$) and negation ($\neg$), but in application, these two connectives are omitted because they are very hard to define and identify [Mooers, 1958].

The main problem in $\mathcal{PL}$ based models is that, they either: **1-** deal with the full spectrum of logical sentences, but in this case they use a very complex inference mechanism for retrieval [Crestani & Rijsbergen, 1995; Picard & Savoy, 2000]. Consequently, these models are inefficient with respect to execution time, where the inference mechanisms are very complex algorithms, or **2-** deal with a restricted category of logical sentences in order to obtain models having an acceptable execution time with respect to inference [Losada & Barreiro, 2001].

IR models that are based on more expressive logics than $\mathcal{PL}$, normally have complex and very hard to obtain document representations e.g. conceptual graph [Chevallet & Chiaramella, 1995], possible world [Nie, 1988; Nie & Brisebois, 1996], etc. Even though, some of them offer very expressive representations.

> **Problem 1:** *In logic-based models, document and query representations are hard to obtain.*

This problem is actually related to the indexing process. In (Figure 1.1), this problem is related to obtaining accurate document representation and query from a document and a request, respectively. We partially consider this issue in the thesis.

[1] The same discussion is also applicable to queries.

1.3.2 Logical Implication

On the one hand, from the beginning, there was a broadly accepted tendency that the classical material implication, denoted $\supset$, is not the correct choice for modeling the retrieval decision $d \rightarrow q$ in logic-based IR models [van Rijsbergen, 1986]. That leaded later to very complex definitions of the implication $d \rightarrow q$. Some researchers even dealt directly with the uncertainty $U(d \rightarrow q)$ without defining what the logical implication $d \rightarrow q$ exactly refers to [Nie, 1988; van Rijsbergen, 1986]. In other words, these studies merge the two steps, retrieval and ranking, in only one step.

On the other hand, different formal logics are used to model the IR implication $d \rightarrow q$. The expressive power of these logics varies from the less expressive (Propositional Logic $\mathcal{PL}$) to a more expressive (First Order Logic $\mathcal{FL}$). However, there is a trade-off between the expressive power of any formal logic and the complexity of its inferencing algorithms. A more expressive logic means more complex inferencing and reasoning algorithms.

The problem is that: when using more expressive logics than $\mathcal{PL}$, the matching between d and q becomes very hard to compute. For example, conceptual graph projection [Chevallet & Chiaramella, 1995], logical imaging (especially when there are a large number of possible worlds) [Crestani & Rijsbergen, 1995], concepts subsumption [Meghini *et al.*, 1993], etc. Even using $\mathcal{PL}$, without restrictions on the logical sentences that could model d and q, leads to a very hard to compute matching [Losada & Barreiro, 2001].

> **Problem 2:** *In logic-based models, matching checking is normally non-operational and non-applicable to large-scale data.*

This problem is related to the matching operation (Figure 1.1).

1.3.3 Uncertainty Definition

In IR literature, researchers use different mathematical theories, including fuzzy logic, probability theory, logical imaging, belief revision, etc., in order to estimate the logical uncertainty $U(d \rightarrow q)$. However, the main disadvantage is that when uncertainty is added to the logic the model rapidly becomes very complex, and the intuitive symmetry between the mathematical model and the studied problem becomes unclear.

Furthermore, the uncertainty measure U is either ad-hoc, e.g. the distance between two possible worlds [Nie, 1988], the cost of changing one conceptual graph to another [Chevallet & Chiaramella, 1995], etc., or it is hard to implement, e.g. probability distributions in logical imaging [Crestani & Rijsbergen, 1995], positioning [Hunter, 1995], etc.

> **Problem 3:** *The uncertainty measure U is either ad-hoc or hard to implement.*

This problem is related to the ranking operation (Figure 1.1).

1.4 Proposed Solution

We propose here a logic-based IR model in order to overcome most of the previous limitations and problems. More precisely, we choose to use Propositional Logic ($\mathcal{PL}$) as a mathematical

framework. However, the novelty in our proposal, comparing to previous logic-based models, is that we exploit the potential relation between logics and lattice theory [Chevallet & Chiaramella, 1995; Knuth, 2005]. Before presenting the solutions that our model offers, we explain why $\mathcal{PL}$ rather than a more expressive logic is used. Simply because, on the one hand, we think that $\mathcal{PL}$ is the most likely logic to build efficient reasoning systems, and then to build logic-based IR models being capable of manipulating large-scale data. On the other hand, even though other logics, like first order logic and description logic, are more expressive than $\mathcal{PL}$ especially in relations representation, it is also possible to represent some simple relations between terms using $\mathcal{PL}$. For example, we can represent the hyponym/hypernym relation through the material implication [Chiaramella & Chevallet, 1992], e.g. $\models dog \supset animal$, $\models pine \supset tree$, etc.

Tackling problem 1

At the level of document and query representation, our model represents d and q as logical sentences written in Disjunctive Normal Form (DNF), and *without any restriction*. Here, we exceed the classical assumption, especially for documents, in logic-based IR models, which says, the logical sentence that represents a document d is the conjunction of the terms that appear in d. For example, assume the vocabulary $A = \{a, b, c\}$, and assume that only the term b appears in d, then according to the classical assumption, d is represented by the following logical sentence $\neg a \wedge b \wedge \neg c$, where in any model of d the terms that appear in d are "true" and all other terms are "false". Whereas, in our model, it is possible to represent d through logical sentences like b which means that d is about b but we do not know if it is about a and c, or like $\neg a \wedge b$ which means that d is about b and excludes a but we do not know if it is about c or not. As we see, it is possible to implicitly represent the uncertainty at the level of indexing. Unfortunately, at the implementation time, we do not deal with the logical negation, because using the $\neg$ operator to describe the content of documents is questionable. Does the $\neg$ operator represent the linguistic negation? For example, assume that the term '*black*' corresponds to a proposition a, then the term '*is not black*' should be represented by $\neg a$ or by another totally different proposition b that corresponds to '*is not black*'. Furthermore, let us forget the language negation, should a document indexed by the color '*red*' directly indexed by $\neg$'*blue*'? Moreover, assume a document indexed by '*Paris*', should this document be indexed by '*France*' or $\neg$'*France*'? knowing that the term '*France*' does not originally appear in the document. Doing such type of reasoning and indexing presupposes that we have the total knowledge to decide about terms, which is a strong assumption. This type of indexing also requires a complete set of rules that make this mutual-exclusion between terms. At the implementation time, we also partially deal with the logical disjunction. However, theoretically, our model is capable of efficiently dealing with any logical sentence, which is not the case in most $\mathcal{PL}$ based IR models. This point represents a partial solution to ***Problem 1***, where there is no restrictions on the logical sentences that our model can efficiently deal with, but it is still hard for some connectives (e.g. $\vee$ and $\neg$) to be automatically identified.

Tackling problem 2

At the level of modeling the logical implication $d \rightarrow q$, we first discuss the possibility to use the material implication to represent the retrieval decision. Then, we exploit the potential

relation between $\mathcal{PL}$ and lattice theory, and also we rewrite logical sentences in a special way, where: **1-** d and q become nodes in a lattice which has a partial order relation equivalent to the validity of material implication, and **2-** transform the implication $d \rightarrow q$ to a series of simple set-inclusion checking. Exploiting the potential relation between $\mathcal{PL}$ and lattices allows us to check the retrieval implication $d \rightarrow q$ in an easy and efficient way, namely a series of set-inclusion checking. This point actually solves ***Problem 2*** in the context of $\mathcal{PL}$, where checking the implication $d \rightarrow q$ becomes easy to verify.

Tackling problem 3

At the level of the uncertainty measure $U(d \rightarrow q)$, since we position d and q on a lattice, and since we transform the material implication between two logical sentences to a partial order relation between their corresponding nodes in the lattice, we suggest exploiting the *degree of inclusion or implication* function Z, which is introduced by Knuth [Knuth, 2005], between two nodes of a lattice, for formally estimating the uncertain implication $U(d \rightarrow q)$. Actually, the degree of inclusion function Z quantifies the partial order relation defined on a lattice, where instead of saying that an element x includes or not another element y, it is possible, using Z, to quantify the degree to which x includes y. Using the degree of inclusion function Z to estimate the uncertainty of an implication allows us to define the uncertainty measure U as an intrinsic part of the logic. Moreover, the function $Z(x, y)$ is exactly the conditional probability $P(x|y)$ if Z is consistent with all properties of distributive lattices [Cox, 1946; Knuth, 2003, 2005], and that actually establishes the connection between our logic-based model and some classical IR models like language models [Ponte & Croft, 1998]. This point actually solves ***Problem 3*** in the context of $\mathcal{PL}$, where the uncertainty $U(d \rightarrow q)$ is directly related to the implication $d \rightarrow q$ through a predefined lattice. Furthermore, in the extreme case, where Z is consistent with all properties of distributive lattices, Z is exactly the conditional probability, which is a very important notion in IR.

To sum up, we propose a logic-based IR model based on $\mathcal{PL}$ as a logical framework. We exploit the potential relation between $\mathcal{PL}$ and lattice theory in order to, on the one hand, transform checking the validity of the logical implication $d \rightarrow q$ to a series of simple set-inclusion checking, on the other hand, exploit the degree of inclusion function Z defined on lattices to estimate the uncertainty $U(d \rightarrow q)$. Finally, our model is capable of working efficiently on any logical sentence without any restrictions, and it is applicable to large-scale data.

1.5 Thesis Structure

This thesis is organized in three main parts: state of the art including chapters 2 & 3, contribution including chapters 4 & 5 & 6, and experiments including chapters 7 & 8. After this general introduction, which describes the general context of this work, its motivation, the problems that need to be solved, and a brief introduction of our contribution, the remaining chapters are organized as follows:

Chapter 2. In this chapter, we talk about the necessity of using a more informative type of terms, namely concepts, and what do studies normally mean by "*concept*". We also review the range of problems that could be solved using concepts, and the range of problems

that could be raised because of using concepts. In addition, the chapter briefly shows some examples of knowledge resources, which contain the concepts and their relations, and it also shows the general paradigm of the text-to-concept mapping process. Finally, the chapter presents some concept-based IR models with their capabilities and shortcomings.

Chapter 3. This chapter starts by reviewing what the symbol '$\rightarrow$' in the implication $d \rightarrow q$ refers to, or in other words, the different statuses of $d \rightarrow q$. The chapter then presents a panoramic overview of most logic-based models in the IR literature, which are organized according to the type of formal logic that is used. The chapter also talks about the two notions Exhaustivity & Specificity. Since in our proposed model we exploit lattices to redefine the implication $d \rightarrow q$ and to compute its uncertainty $U(d \rightarrow q)$, then at the end of this chapter, we present some examples of using lattice theory in IR, especially the Formal Concept Analysis (FCA) technique. Actually, lattices are used in IR in a very different way from ours.

Chapter 4. We mainly address in this chapter *Problem 2*, where we discuss in detail the widely-accepted assumption about the need for a non-classical implication to represent the IR retrieval decision, and that the material implication is not suitable for IR. We show the inadequacy of this assumption through showing the inconvenient argumentation under it. This chapter ends with a new hypothesis stating that the validity of material implication $\models d \supset q$ is a suitable choice to represent the IR retrieval decision.

Chapter 5. We address in this chapter *Problems 1 & 2 & 3*, where we present our logic and lattice based IR model. We start by presenting the mathematical connection that we establish between $\mathcal{PL}$ and lattice theory. The connection is based on re-expressing logical sentences in a different way, and then each logical sentence becomes a set of nodes in a predefined lattice. Accordingly, we transform the checking of material implication validity $\models d \supset q$ to a series of simple set-inclusion checking. After that, the chapter presents our logic-based IR model, which represents documents and queries as logical sentences without any restriction, and the retrieval decision $d \rightarrow q$ as the validity of material implication $\models d \supset q$. For estimating the logical uncertainty $U(d \rightarrow q)$, we exploit the degree of inclusion function Z, which is already introduced and defined on lattices. Finally, the chapter discusses some direct conclusions of our model, including, formalizing and showing the adequacy of van Rijsbergen assumption about estimating the logical uncertainty $U(d \rightarrow q)$ through the conditional probability $P(q|d)$, redefining the two notions Exhaustivity & Specificity, and the possibility of reproducing most classical IR models as instances of our model.

Chapter 6. We link in this chapter our proposed model with *Motivations 1 & 2*, where we move from our theoretical model, proposed in the previous chapter, to more operational aspects. We explain the mapping between indexing terms and atomic propositions, and we also demonstrate what the truth and falseness of indexing terms refer to. We present in this chapter three instances of our model: 1- basic instance, which shows the importance of integrating Exhaustivity & Specificity into IR models, 2- relation-based instance, which shows the advantages of exploiting semantic relations between terms, and 3- structure-based instance, which exploits the potential structure within the text.

Chapter 7. In this chapter, we show our experimental setup, including, the corpora that are used, the types of indexing terms, the way of extracting these terms from the text, term

weighting schemes, and some baseline results.

Chapter 8. This chapter is mainly dedicated for the experimental results of the instances of our model, and also for discussing the results that we obtain with respect to baselines.

Chapter 9. This chapter includes the general conclusions and the main perspectives of this thesis.

Appendix A. This appendix presents our contribution in the knowledge-based IR. We mainly talk about our new approach of concept counting, namely the Relative Concept Frequency (RCF). RCF exceeds the flat representation of documents and queries through exploiting some structural relations. RCF forms the basis of the weighting schema in the structure-based instance of our model.

Appendix B. In this appendix, we recall some mathematical definitions and properties of lattices. We also talk about the degree of inclusion function Z and some of its interesting properties.

Appendix C. This appendix reviews some definitions and theories related to $\mathcal{PL}$. The appendix mainly focuses on the formal interpretation of $\mathcal{PL}$.

In case that the reader is not familiar with the following mathematical notions: lattice theory, quantifying the lattice-related partial order relations, and the formal language and semantic of $\mathcal{PL}$, it is preferable to read (Appendices B&C) before continuing reading this thesis.

Part II

STATE OF THE ART

Chapter 2

Knowledge-Based IR Models

2.1 Introduction

When somebody says that a document d is relevant to a query q, he/she implicitly brings his/her background knowledge to do this judgment. Therefore, if d is relevant to q according to a user u_1, that does not forcibly mean, d is relevant to q according to another user u_2, because users have different background knowledge and needs. In addition, the same query q is evaluated differently according to the field of study. For example, the meaning of '*x-ray*' in physics is different from its meaning in medicine. Therefore, there is always an external factor affecting the IR process, this factor is a type of knowledge related to users who ask the query and to the field in which the query is evaluated.

Besides that, the assumption about the independence of terms is not totally true, because terms are normally connected to each other via some linguistic and semantic relations, e.g. synonymy, antonymy (opposition), hyponymy-hypernymy (specific-general), meronymy-holonymy (part-whole), etc. For example, in order to establish a matching between a document containing the term '*cancer*' and a query containing the term '*malignant neoplastic disease*', we need a knowledge resource containing information about these two terms and that they are synonymous.

Knowledge-based IR models are the models that explicitly exploit external[1] resources of knowledge in order to build a more precise representation of documents and queries (knowledge-based indexing), or to build a system's relevance judgment closer to the human way of relevance judgment (knowledge-based matching).

Knowledge is organized in external resources, e.g. UMLS[2], WordNet[3], DBpedia[4], etc. There are several ways to organize knowledge into resources. One of the simplest ways is to represent knowledge as a set of elements connected via some relations [Baader *et al.*, 2003; Davis *et al.*, 1993; Meghini *et al.*, 1993; Ounis & Huibers, 1997]. A knowledge resource K can thus be represented by the structure $K = \langle V, R \rangle$, where V is the vocabulary and it is a set of elements, and $R = \{r | r \subseteq V \times V\}$ is a set of relations between vocabulary elements. Concerning the knowledge resources used in IR, their vocabularies range from simple words to

[1] External with respect to documents and queries.

[2] Unified Medical Language System (http://www.nlm.nih.gov/research/umls/).

[3] WordNet is a lexical database of English (http://wordnet.princeton.edu/).

[4] DBpedia is a data set derived from Wikipedia (http://dbpedia.org).

some more abstract and sophisticated elements as concepts[1]. Relations also range from simple linguistic relations, e.g. synonymy, to some conceptual and semantic relations, e.g. a type of bacterium *causes* a type of infection.

After talking in general about the importance of knowledge resources in IR, in this chapter, we mainly focus on concepts and the relations between them, since concepts and their relations form a common way of knowledge organization and representation. We also present how concepts and relations are used to build IR models. In general, concepts are supposed to be more informative terms than simple words [Chevallet *et al.*, 2007].

We start this chapter by presenting different definitions of concepts. In section 3, we mention some general motivations of using concepts instead of the simpler type of terms, namely words. In the same section, we also show the range of problems that can be solved using concepts, and the range of problems that come to light because of concepts. In section 4, we present two examples of knowledge resources and their internal structure, and we also present the way that is used to annotate a chunk of text by concepts. We briefly introduce two tools that achieve this type of annotation or mapping. In section 5, we review three IR models originally proposed to deal with concepts and to exploit inter-concepts relations. We conclude in section 6.

2.2 What is a Concept?

Philosophically, concepts are the basic units of human thought. In psychology, a concept X is the part of knowledge that is used in some cognitive processes (e.g. categorization, induction, etc.) requiring the concept X. Machery in ([Bělohlávek & Klir, 2011], Chapter 2) says:

> "*The concept of dog is a subset of knowledge about dogs. It is retrieved from long-term memory in a context-insensitive manner, and it is used in the process underwriting our higher cognitive competences. We use it to decide whether to classify something as a dog, to make inductions about dogs, to understand sentences containing the word 'dog', and so forth.*"

In his tutorial ([Bělohlávek & Klir, 2011], Chapter 2) about concepts, Machery reviews the main four psychological theories of concepts:

Classical theories. In these theories, a concept is described through a set of properties or through a Boolean expression of these properties.

> "*A concept of X represents some properties as being separately necessary and jointly sufficient to be an X.*"

For example, someone is *Giant* if and only if he/she is a human, an adult, and his/her tall is more than a predefined threshold.

> "*A concept of X can consist of a representation of any Boolean combination of properties provided that this combination states a necessary and sufficient condition for being an X.*"

[1] A word is the smallest linguistic element that has a semantic and can stand by itself. We present what a concept means in (Section 2.2).

For example, someone is *very short* if and only if (he/she is a child) OR (he/she is an adult AND his/her tall is less than a predefined threshold).

Prototype theories. In these theories, a concept is defined through some classes of properties.

> "*A concept of X represents either the properties that are typical of category members, the properties that are diagnostic of them, or the properties that best weigh typically and diagnostically. A property is typical if the probability that a particular possesses this property if it belongs to the category is high, whereas a property is diagnostic if the probability that a particular belongs to the category if it possesses this property is high.*"

Table 2.1 in [Bělohlávek & Klir, 2011] presents a possible prototype of the concept *vehicle*, where a *vehicle carries people or things*, *can move*, *has wheels*, etc.

Exemplar theories. In these theories, concepts are described through a set of exemplars, or simply a concept is an abstraction of a set of exemplars or objects.

> "*A concept of dogs consists in a set of representations of particular dogs (say, a representation of Fido, a representation of Rover, etc.), which are used in the cognitive processes underlying our higher cognitive competences.*"

For example, the concept *President* is the set of all presidents all over the world.

Theory theories. In these theories, a concept is manipulated as a scientific theory.

> "*Like scientific theories, concepts consist of knowledge that can be used to explain events, phenomena, or states of affairs. Theory theorists hold that casual knowledge, nomological*[1] *knowledge, functional knowledge, and generic knowledge are all used in explanation, and, thus, that concepts consist of these types of knowledge.*"

For example, a theory of *Human* could be a body of causal and functional knowledge, where causal knowledge like a human cries because he/she is sad or happy, and functional knowledge like a human sweats to cold his/her body.

Others. Chevallet et al. [Chevallet *et al.*, 2007] present another definition of concepts:

> "*Concepts can be defined as a human understandable unique abstract notions independent from any direct material support, independent from any language of information representation, and used to organize perception and knowledge.*"

[1] Nomological: relating to or denoting principles that resemble laws, especially those laws of nature which are neither logically necessary nor theoretically explicable, but just are so. [Online Oxford Dictionaries (www.oxforddictionaries.com)]

In computer science, concepts are defined in a simpler manner. Meghini et al. [Meghini *et al.*, 1993] and Wille [Wille, 1982] define a concept as a category or a class that is described either through a set of properties or attributes, or through the set of objects or instances belonging to it. This definition of concepts merges: the classic theory and the exemplar theory of concepts. In IR, sometimes, the term '*concept*' is also used for referring to words or phrases, where each word or phrase is a possible concept [Bendersky *et al.*, 2011].

In this study, we see concepts in a very close manner to their representation in knowledge resources like UMLS or WordNet, and also in a very close manner to the exemplar theory, where concepts in this theory are statistical notions.

Definition 2.1 (Concept)**.** *A concept is the identifier of the set that encompasses synonymous phrases or words. Concepts are usually entries in a knowledge resource.* □

For example, the two synonymous phrases '*Atrial Fibrillation*' and '*Auricular Fibrillation*' belong to the same concept '*C0004238*' in UMLS. The two phrases '*cancer*' and '*malignant neoplastic disease*' belong to the same synset in WordNet.

2.3 Why Concepts?

Words have been used for a long time in IR, and this type of indexing terms proved its effectiveness in most IR applications, especially web search engines. Besides the reasons that are mentioned in the introduction, using concepts is motivated by some other reasons.

A rich and large knowledge resources, which are considered the main containers of concepts, are now available, e.g. UMLS, WordNet, etc. Concepts also allow us to deal with some special issues of multilingual and multi-modal content [Chevallet *et al.*, 2007; Ren & Bracewell, 2009]. For example, it is possible to abandon the translation step in a multilingual context, because concepts are supposed to be language-independent, e.g. the English word '*lung*' and the French word '*poumon*' correspond to the same concept '*C0024109*' in a knowledge resource like UMLS. Moreover, some new semantic-based IR applications, e.g. Semantic Web [Ren & Bracewell, 2009] and song indexing and retrieval [Codocedo *et al.*, 2012], require a more sophisticated way of representation and reasoning.

Concepts also contribute to solve some well-known IR problems like the term-mismatch problem [Crestani, 2000]. This problem happens when using two different terms to express the same meaning, e.g. '*atrial*' vs. '*auricular*'. In the ideal case, each concept should encompass all terms that have the same meaning in a specific context. Therefore, replacing words and phrases by their corresponding concepts contributes to partially solve the term-mismatch problem. As term-mismatch is a very important problem in IR, we devote a separated section to describe the problem and how concepts could help to overcome this problem.

IR is an inferential process [Chiaramella & Chevallet, 1992; van Rijsbergen, 1986]. In fact, concepts fit well this inferential nature of IR, because concepts are normally elements of a human-verified knowledge resource, and they are also linked together through some predefined semantic relations. More precisely, concepts are usually accompanied by a rich source of information that helps to simulate the inferential nature of IR process.

All reasons and motivations above lead to the emergence of an IR field that uses concepts as indexing terms instead of, or besides, words.

2.3.1 Term-Mismatch Problem

> "*How often have you tried to look up something in an index and failed to find what you were looking for because the words or phrases you looked for were different from those used in the material you needed to find?*" [Woods, 1997]

In natural languages, there are many ways to express the same meaning, or equivalently, two terms could have the same meaning in a specific context. For example, '*atrial*' vs. '*auricular*', '*apartment*' vs. '*flat*', '*air pollution*' vs. '*pollution of the air*', etc. This is one of the features of natural languages that give each author the ability to have her/his own writing style. However, in IR field, it is a problematic feature, because most IR systems use a type of query-document intersection. Therefore, by using different terms, in queries and documents, for expressing the same meaning, IR systems will not be able to retrieve relevant documents. This problem is well studied in IR literature and is called *term-mismatch* problem [Chevallet, 2009; Crestani, 2000]. More precisely, if a user in his/her query uses a term different from the term that is used by the author of a document to express the same thing, in this case, by simple term intersection between document's terms and query's terms, the system cannot retrieve the document. Therefore, without an external knowledge resource, that links synonymous terms, the system cannot retrieve, for example, a document containing '*apartment*' as a response to a query containing '*flat*'. According to Crestani [Crestani, 2000], in less than 20% of cases, two people use the same term to describe the same meaning.

Some researchers use phrases instead of words as indexing terms [Bendersky *et al.*, 2011; Ho *et al.*, 2006]. They suppose that a phrase is more precise and informative than individual words. Term mismatch problem also extends to phrases' level. For example, the following two phrases '*skin cancer*' and '*melanoma*' have a close meaning. At phrases' level, term mismatch problem is also related to phrase variations, e.g. '*air pollution*' and '*pollution of the air*'.

The term-mismatch problem was heavily studied by several researchers. In IR literature, several approaches, to solve this problem, could be identified.

Query expansion. This approach expands the query using some new terms to increase the chance of matching with documents [Efthimiadis, 1996]. Normally, a query is expanded by the synonymous terms of the query's original terms. Queries could also be expanded by applying the pseudo relevance feedback technique, which chooses some terms from the top ranked documents [Buckley *et al.*, 1994; Rocchio, 1971; Salton & Buckley, 1997].

Using term-term semantic similarity measures. This approach presupposes the existence of a measure being capable of estimating the similarity between any two terms [Crestani, 2000; Qiu & Frei, 1993].

$$\forall t_i, t_j \in T, \quad 0 \leq Sim\,(t_i, t_j) \leq 1$$

where T is a set of terms. Using this measure, the matching score (or the Relevance Status Value RSV) between a document d and a query q can be computed even if they do not share any term.

$$RSV(d, q) = \sum_{t \in q} Sim(t, t^*) \times w_d(t^*) \times w_q(t)$$

where, $t^* \in T$ is the most similar document term to the query term t, $w_d(t^*)$ is the weight of the term t^* in d, and $w_q(t)$ is the weight of the term t in q.

There are many semantic similarity measures. Some of them are applicable whatever is the type of terms[1] [Chevallet, 2009; Qiu & Frei, 1993], but others depend on the type of terms and the inter-terms structure [Aslam & Frost, 2003; Holi & Hyvönen, 2005; Li *et al.*, 2003; Mohler & Mihalcea, 2009].

Dimensionality reduction. This approach reduces the chance that a query and a document use different terms for representing the same meaning. Among the techniques that are used for achieving this mission, we can mention: Stemming [Frakes, 1992], Latent Semantic Indexing (LSI) [Deerwester, 1988; Deerwester *et al.*, 1990], and Conceptual Indexing (using concepts instead of words or phrases) [Chevallet *et al.*, 2007]. Here, we can see the importance of concepts in solving the term-mismatch problem, where each concept normally encompasses a set of synonymous words and phrases (Definition 2.1). We focus in the next section on the concept-based solution of the term-mismatch problem.

2.3.2 Concepts to Solve Term-Mismatch

To solve the term-mismatch problem, many researchers proposed to use concepts as indexing terms. Assume the two synonymous terms t_1 and t_2, which correspond to the same concept c. If the content of a document d is described using the term t_1 and a query q is asked using the term t_2, then in this case, we get a mismatch between d and q. Whereas, if we replace the two terms t_1 and t_2 by their corresponding concept c, then d and q will be described using the same concept c. For example, the two phrases '*Atrial Fibrillation*' and '*Auricular Fibrillation*' correspond to the same concept '*C0004238*' in UMLS.

However, using concepts *partially* solves the term-mismatch problem, because sometimes two related terms t_1 and t_2 correspond to two different concepts c_1 and c_2 without having a relation between c_1 and c_2. We called this problem *concept-mismatch*. It normally results from the inconsistency and incompleteness of knowledge resources [Bodenreider *et al.*, 1998, 2001]. For example[2], the two terms '*B-Cell*' and '*Lymphocyte*' correspond to the two concepts '*C0004561*' and '*C0024264*', respectively, but there is a relation of type '*ISA*' between the two concepts. Whereas, the two terms '*Dermatofibroma*' and '*Dermatofibrosarcoma*'[3] correspond to two different concepts '*C0002991*' and '*C0392784*', respectively, and there is no relation linking these two concepts, where the concepts should be related because they refer to two diseases affection the same body-part and with very close symptoms.

2.3.3 Problems Caused by Concepts & State of the Art Solutions

Besides the concept-mismatch problem, using concepts as indexing terms poses some other problems. Using concepts requires a new indispensable operation for mapping textual content

[1] In IR, many types of terms could be used to index documents and queries, e.g. words, concepts, phrases, ngram of characters, etc.

[2] The two examples are extracted from UMLS.

[3] '*Dermatofibroma*' and '*Dermatofibrosarcoma*' are benign tumor and malignant tumor, respectively, affecting the skin with very similar symptoms.

Table 2.1: The different meanings of '*x-ray*' in UMLS

Meaning	Corresponding concept
Roentgenographic (functional concept)	*C0034571*
Diagnostic radiologic examination (diagnostic procedure)	*C0043299*
Roentgen rays (natural phenomena or process)	*C0043309*
Plain x-ray (diagnostic procedure)	*C1306645*
Clinical attribute	*C1714805*
Radiographic imaging procedure (diagnostic procedure)	*C1962945*

to concepts. It is a complex, imperfect, and time-consuming operation. Moreover, the mapping operation forms only one step of the global conceptual indexing process. The main principle of mapping is to identify noun phrases, and then to try to map those noun phrases to concepts of a knowledge resource [Aronson, 2006; Chevallet *et al.*, 2007; Dozier *et al.*, 2007; Maisonnasse *et al.*, 2009]. In addition to the imperfection of mapping tools, knowledge resources, which contain concepts, are generally incomplete or non-exhaustive [Bodenreider *et al.*, 1998, 2001]. For example, the term '*Osteoporotic*' does not map to any concept in UMLS (version 2012AA).

Furthermore, the retrieval performance of any concept-based IR model is highly depended on: 1- the quality and the completeness of the knowledge resources, which contain concepts, 2- the precision of the text-concepts mapping process, and 3- the amount of information that is used besides concepts, such as the relations between concepts. In other words, to which degree we benefit from the content of knowledge resources. Actually, there is a trade-off between the amount of information that is used and the simplicity and applicability of the model.

Since natural languages are ambiguous, a word or a phrase could have several meanings, and corresponds to several concepts. For example, '*x-ray*' is mapped to six different concepts in a knowledge resource like UMLS (Table 2.1). Therefore, we need an additional conceptual disambiguation step, which chooses, among the candidate concepts, the concept that best fits the underlying context. *Conceptual Disambiguation* is defined as choosing the most appropriate concept, among the candidate concepts, that best corresponds to the related context. Since a concept that corresponds to a term could be seen as a possible meaning or sense of that term, then it is possible to generalize the conclusions of the classical Word Sense Disambiguation (WSD) to the conceptual disambiguation. The main conclusion of applying a sort of disambiguation to IR is that disambiguation processes must be very accurate (about 90%) in order to slightly improve the IR systems performance (about 4%) [Navigli, 2009; Sanderson, 1994]. For more information about disambiguation processes, Navigili [Navigli, 2009] reviews the main approaches of Word Sense Disambiguation (WSD) in the Artificial Intelligence (AI) field. In addition, Sanderson [Sanderson, 1994] and Baziz [Baziz, 2005] review the main disambiguation techniques that are used in IR.

Besides that, moving from the word-space to the concept-space changes the classical notion of term frequency. More precisely, the general assumption in the word-based IR models is that if a document d contains two words w_1 and w_2 then d should be represented by the meaning of w_1 *AND* the meaning of w_2. However, this is not the case in the concept-based IR models, because a word w in a document d is normally mapped to a set of concepts $\{c_1, \ldots, c_n\}$, where each concept represents a possible meaning of w, then d should be represented by one of these

meanings or concepts that best corresponds to the content of d. In other words, d should be represented by c_1 *OR* c_2 *OR* ... *OR* c_n. Accordingly, document and query lengths change in a *non-consistent* way when moving from the word-space to the concept-space.

Concerning the concept-mismatch problem, many approaches exist in literature. Exploiting semantic relations between concepts, especially the hyponymy/hypernymy relation, could alleviate the concept-mismatch problem [Baziz, 2005; Le, 2009; Maisonnasse, 2008], e.g. exploiting the *ISA* relation between the two concepts that correspond to the two terms '*B-Cell*' and '*Lymphocyte*'. Concept-based query expansion, or in other words, expanding queries by concepts rather than words, could also alleviate this problem [Aronson & Rindflesch, 1997; Baziz, 2005]. Indexing documents and queries by *domain dimensions*, which are more abstract elements than concepts, could also contribute to solve this problem [Radhouani, 2008].

One of the candidate techniques that could also contribute to solve the concept-mismatch problem is *data fusion*, because this technique allows to compensate the lack of information that causes mismatch by other sources of information. In IR, for a certain document collection (corpus), *data fusion* is the process of combining different result sets of a certain query (information need) [Vogt & Cottrell, 1998, 1999]. A result set of a query is a list of documents, ordered according to their expected relevance score. Actually, different result sets of the same query, in the same corpus, could be produced through [Croft, 2000]: using totally different IR systems [Fox & Shaw, 1994; Shaw *et al.*, 1994], or using the same IR system but with different configurations. Theses configurations include: 1- building different representations of documents and queries using different types of indexing terms [Bartell *et al.*, 1994; Baziz, 2005; Das-Gupta & Katzer, 1983], different parts of documents [Das-Gupta & Katzer, 1983], different weighting schema [Lee, 1995], etc., 2- describing the same information need by different queries [Belkin *et al.*, 1993, 1995], and 3- using different ranking algorithms (matching formulas) [Fox & Shaw, 1994; Shaw *et al.*, 1994]. Hence, to compensate the mismatch at the level of concepts, it is possible for example, based on data fusion techniques, to combine concepts with words, either using a late or early fusion technique [Baziz, 2005].

2.4 Conceptual Mapping and Indexing

Conceptual mapping is the process of mapping text to concepts of a predetermined knowledge resource. The main principle of conceptual mapping process is to extract *phrases* from the text of documents and queries, and then try to map them to one or more *candidate concepts* from a knowledge resource. More precisely, the general process of conceptual mapping consists of the following steps [Chevallet *et al.*, 2007]:

1. *Morphology and syntax*: extracting noun phrases from text.
2. *Variation*: constructing a list of variants for each noun phrase. Variants could be derivational variants, synonyms, acronyms, etc.
3. *Identification*: for each variant, all concepts that could correspond to it are retrieved from the knowledge resource. The retrieved concepts called *candidate concepts*.
4. *Evaluation*: for each candidate concept, a measure is used for evaluating the precision of mapping process, and then the set of candidate concepts is ordered according to this

measure. In other words, the measure computes the degree of correctness of mapping a noun-phrase to a concept.

5. *Disambiguation*: choosing the most appropriate concepts, among the candidate concepts, that well correspond to the related noun-phrase. This operation normally depends on the context.
6. *Weighting*: like in word-based indexing, each concept has a weight reflecting its indexing usefulness.

Conceptual indexing is the process of transforming the content of documents and queries from its original form (e.g. text), to a predefined concept-based representation (e.g. graph of concepts). Conceptual indexing first maps text to concepts. When the mapping is done, the indexing process must continue by first selecting and sometimes weighting the concepts, and then representing documents and queries. The system is then able to achieve the concept-based matching between a query and a document.

Actually, there are many examples of mapping tools. MetaMap [Aronson, 2006], for example, maps medical text to UMLS concepts. Fast Tagging [Dozier *et al.*, 2007] is a method of tagging medical terms in legal, medical, and news text, and then mapping the tagged terms to UMLS concepts. Baziz [Baziz, 2005] and Maisonnasse [Maisonnasse, 2008] built their own mapping tools. Maisonnasse [Maisonnasse *et al.*, 2009] studied the effect, on retrieval performance, of merging the output of several mapping tools.

In general, concepts are a part of a knowledge resource, it is thus mandatory to link mapping tools to some knowledge resources, e.g. UMLS, WordNet, DBpedia, etc. In the following subsections, we present examples of knowledge resources and mapping tools.

2.4.1 Knowledge Resources

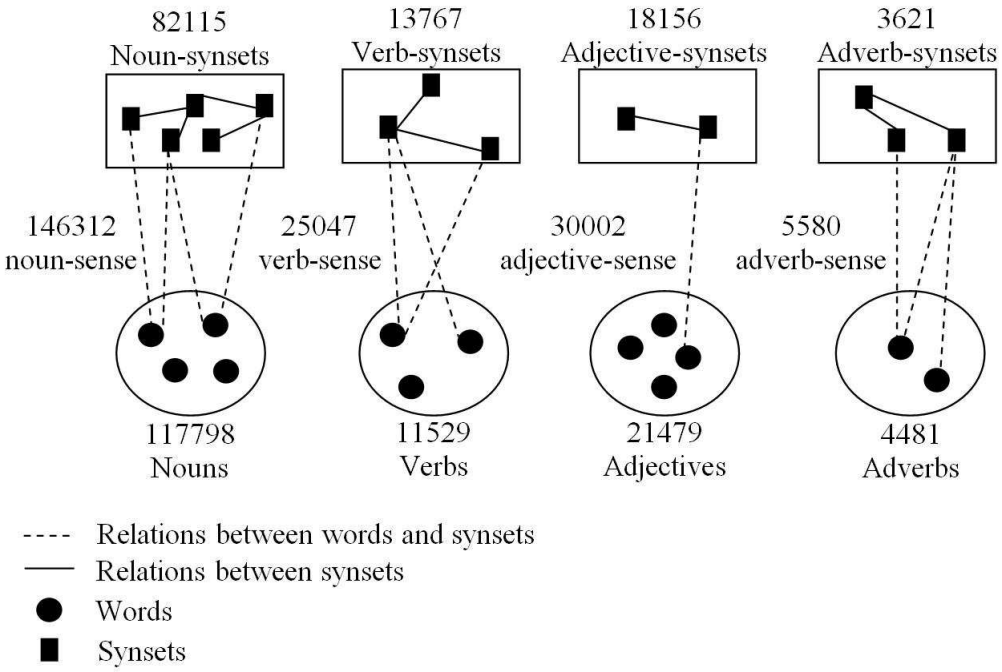
In this section, we talk about two knowledge resources: *WordNet* which is a general purposes lexical resource, and *UMLS* which is a meta-thesaurus in the medical domain. Of course, there are many other resources, e.g. Open Directory Project (ODP)[1], Yet Another Great Ontology (YAGO)[2], etc. There are also resources used in other disciplines. For example, British National Corpus (BNC)[3] in Natural Language Processing. FOAF[4] (from "friend of a friend"), Semantically-Interlinked Online Communities Project (SIOC)[5], etc., in Semantic Web. Resources in Semantic Web are normally formal and based on Description Logic. We choose to describe WordNet and UMLS, because they are often used in IR and also we use UMLS in our experiments.

2.4.1.1 WordNet

WordNet is a lexical database of English, developed at Princeton University. English vocabularies are grouped into sets of cognitive synonyms (*synsets*). In other words, synonyms are grouped

[1] www.dmoz.org
[2] www.mpi-inf.mpg.de/yago-naga/yago/
[3] www.natcorp.ox.ac.uk
[4] xmlns.com/foaf/spec/
[5] sioc-project.org/ontology

Figure 2.1: WordNet



together in one synset. Synsets are linked together through a set of lexical and conceptual-semantic relations. WordNet categorizes English vocabulary into four categories: *Nouns*, *Verbs*, *Adjectives*, and *Adverbs*.

Each synset has a brief description (*gloss*), and it corresponds to a specific meaning (*sense*). Accordingly, each word could be a member of multiple synsets because it could have multiple senses according to the context. Words inside the same synset implicitly have a synonym relation. Synsets are linked together via some relations, e.g. antonymy (opposition), hyponymy-hypernymy (specific-general), meronymy-holonymy (part-whole).

WordNet 3.0[1] database contains about $155K$ unique strings distributed on about $118K$ synsets. It also contains about $207K$ word-sense pairs, see (Figure 2.1). For comparison, Oxford English Dictionary contains about $200K$ English words[2].

2.4.1.2 Unified Medical Language System (UMLS)

UMLS is a multi-source meta-thesaurus in the medical domain, and it contains three main components:

Meta-thesaurus. Meta-thesaurus is a vocabulary database in the medical domain, extracted from many sources; each source of them is called *Source Vocabulary*. Meta-thesaurus is organized into concepts, which represent the common meaning of a set of strings extracted from

[1]This statistic is extracted on 25/10/2013 from the following web page:
http://wordnet.princeton.edu/wordnet/man/wnstats.7WN.html

[2]http://www.oxforddictionaries.com/words/how-many-words-are-there-in-the-english-language (consulted on 19/12/2013)

Figure 2.2: UMLS

Concept (CUI)	Terms (LUIs)	Strings (SUIs)	Atoms (AUIs) * RRF Only
C0004238 Atrial Fibrillation (preferred) Atrial Fibrillations Auricular Fibrillation Auricular Fibrillations	**L0004238** Atrial Fibrillation (preferred) Atrial Fibrillations	**S0016668** Atrial Fibrillation (preferred)	**A0027665** Atrial Fibrillation (from MSH) **A0027667** Atrial Fibrillation (from PSY)
		S0016669 (plural variant) Atrial Fibrillations	**A0027668** Atrial Fibrillations (from MSH)
	L0004327 (synonym) Auricular Fibrillation Auricular Fibrillations	**S0016899** Auricular Fibrillation (preferred)	**A0027930** Auricular Fibrillation (from PSY)
		S0016900 (plural variant) Auricular Fibrillations	**A0027932** Auricular Fibrillations (from MSH)

different source vocabularies. These concepts are linked together via a variety of relations. There is a specific structure linking concepts to their sources vocabulary. The structure encompasses (Figure 2.2):

- Strings: represent the different forms of the same concept. A concept is a meaning; A meaning can have many different names.
- Atoms: The same string may appear in different sources vocabulary, so an atom is a specific string in a given source vocabulary.
- Terms: It is possible that different strings are lexical variants of each other, so these strings are linked to the same term. An UMLS *term* is the group of all strings that are lexical variants of each other.

UMLS is reasonably big resource in the medical field, where the meta-thesaurus $2011AA$ release contains about $2405K$ concepts, $7955K$ terms, $8846K$ strings, and $10655K$ atoms. This content comes from 134 distinct sources and distributed on 21 different languages.

Semantic Network. Semantic Network contains a set of *Semantic Types* linked together via two different types of *Semantic Relations*:

- A hierarchical relation (*ISA* relation).
- A set of non-hierarchical relations: UMLS groups non-hierarchical relations into five main categories: '*physically related to*', '*spatially related to*', '*temporally related to*', '*functionally related to*', and '*conceptually related to*'.

Semantic Network contains 133 Semantic Types and 54 Semantic Relations. The purpose of Semantic Network is to provide a consistent categorization of all concepts in UMLS Metathesaurus.

SPECIALIST Lexicon and Lexical tools. SPECIALIST Lexicon is a set of general English or biomedical terms and words extracted from different sources. Each entry in the Lexicon is a record called *unit lexical record*, which contains a list of lexical information about the related term or word.

Concerning Lexical tools, the goal of these tools is to obtain the base form of a term or word. In other words, these tools are used to abstract a word from any lexical extensions. For example, these tools transform the three strings '*melenoma*', '*melenomon*', and '*melenomun*' to one normalized form '*melenoma*'.

2.4.2 Mapping Tools

In this section, we present two examples of mapping tools. MetaMap [Aronson, 2006], which uses NLP (Natural Language Processing) techniques for recognizing noun phrases, and then tries to map them to UMLS concepts. Fast tagging [Dozier *et al.*, 2007], which uses very efficient method for tagging medical terms (UMLS concepts) within legal, medical, and news text. Actually, there are other methods, see for example the method that is used in [Baziz, 2005] to map text to WordNet synsets. See also [Maisonnasse, 2008; Maisonnasse *et al.*, 2009] that study several mapping tools and their effects on the retrieval performance.

2.4.2.1 MetaMap

MetaMap is a tool of mapping medical text to UMLS concepts. The whole process of MetaMap with all technical details is clarified in [Aronson, 2006]. According to MetaMap, text is a set of utterances U.

$$U = \{u_i | u_i \quad \text{is an utterance}\}$$

The first step in the mapping process is to parse each utterance into a set of noun-phrases using the *SPECIALIST* tagger and the *MedPost/SKR* part of speech tagger.

$$\forall u_i \in U, u_i = \{p_{ij} | p_{ij} \quad \text{is a noun-phrase identified in } u_i\}$$

The second step is to generate variants for each noun-phrase p_{ij}. A variant is a meaningful sequence of one or more words of p_{ij}, with all synonymous, abbreviations, acronyms, spelling, derivational, and inflectional variants. Each variant has its variant distance score that measures the degree to which this variant varies from its original noun-phrase.

$$\forall p_{ij} \in u_i, \mathit{Var}_{ij} = \{(v_{ij}^k, \mathit{dist})\}$$

where, v_{ij}^k is a variant of p_{ij}, dist is the distance between p_{ij} and v_{ij}^k. The third step is to identify the Meta-thesaurus candidate concepts for each noun-phrase, where each Meta-thesaurus concept containing one of the variants of a noun-phrase is a candidate concept for that noun-phrase.

$$\forall p_{ij} \in u_i, CC_{ij} = \{c_l | c_l \in M, \exists v_{ij}^k \in \mathit{Var}_{ij}, v_{ij}^k \in c_l\}$$

where, M represents the set of concepts of UMLS Meta-thesaurus, CC_{ij} is the candidate set of concepts of the noun-phrase p_{ij}, and c_l is a concept in M and is considered as a set of strings. The fourth step is evaluating the precision of the mapping between a noun-phrase and a candidate concept, and then candidate concepts are ordered according to this evaluation function. For detailed information about the evaluation function, please refer to [Aronson, 2006]. The overall evaluation value is normalized to a value between 0 (no match at all) and 1000 (identical match).

The fifth step is to reduce the size of the candidate set of concepts, where for each subset of candidate concepts, which correspond to the same part of the original noun-phrase, the best candidate concept, according to the evaluation function, is chosen.

2.4.2.2 Fast Tagging

Fast tagging [Dozier *et al.*, 2007] is a method of tagging medical terms in legal, medical, and news text, and then mapping the tagged terms to UMLS concepts. The method depends on finding the longest sequence of continuous words that corresponds to a medical term in a predefined authority file, and then converting the tagged term to a hash key to retrieve the corresponding UMLS concepts.

The tagging process starts by building an authority file containing the medical terms that are extracted from UMLS Meta-thesaurus and Red Book drug reference database, and then categorizing them into five categories: injuries, diseases, medical procedures, medical devices, and drugs.

For efficiency purposes, each medical term is assigned a hash key calculated according to specific rules. Each medical term is also assigned one of three ambiguity levels: 1) *unambiguous*: if it is always used in a medical sense, even in non-medical text, 2) *ambiguous*: if it is sometimes used in a non-medical sense in non-medical text, or 3) *problematic*: if it is rarely used in a medical sense in non-medical text. In medical text, the system tags the three types of medical terms. However, in non-medical text, the system ignores the problematic terms and tags the ambiguous terms only if they have a medical context.

The first two steps, building the authority file and determining the ambiguity level, are achieved offline before the actual tagging process starts. After the first two offline steps, the system tries to find the longest sequence of words in text that could be converted to a hash key corresponding to one of keys in the authority file. In this case, this sequence is tagged as a medical term and assigned the UMLS concepts IDs that have the same hash key value.

The last step is to determine the correct sense of ambiguous terms according to their context (words before and after the term). The main advantage of this method is its efficiency.

2.4.3 Conclusion

We presented two examples of knowledge resources (UMLS and WordNet), and also two examples of mapping tools (MetaMap and Fast Tagging). The goal of this presentation is two-fold. First, we tried to give an idea about what these tools and resources actually look like, and how they are built. Second, we tried to show that knowledge resources are not exhaustive, where it is almost impossible to build an exhaustive and complete resource because completeness means that we have total knowledge, which is a strong assumption. Furthermore, we tried to show that

mapping tools are not perfect because they use some approximation techniques for text annotation and concept mapping. These two conclusions, namely knowledge resources are incomplete and mapping tools are not perfect, represent the main two problems facing concept-based IR.

2.5 Concept-Based IR

In general, there are two ways to use external knowledge resources, which encompass concepts and their relations, in IR systems, either *partial* use (query expansion), or *extended* use (in both indexing and matching). External resources are also used in IR-related fields like text classification [Albitar, 2013].

There is a large number of studies that partially integrate an external knowledge resource in IR process. These studies differ either in the used resource, or in the way of using this resource.

Some studies index documents and queries based on external resource, but they use a classical IR model for retrieval. Voorhees [Voorhees, 1993, 1994] and Gonzalo et al. [Gonzalo *et al.*, 1998] use the traditional vector space model, but instead of indexing documents and queries using words, they index them using WordNet's synsets. Vallet et al. [Vallet *et al.*, 2005] also use vector space model for retrieval, but they semi-automatically annotate documents and queries using classes of some taxonomies. Zhou et al. [Zhou *et al.*, 2006] index documents and queries using UMLS concepts, and they use classical language models for retrieval. Diem Le et al. [Diem *et al.*, 2007] apply DFR (Divergence From Randomness) like weighting to documents and queries indexed by UMLS concepts.

The other way of partial external-knowledge-resources integration is the query and document expansion, where external resources are exploited to enrich documents and queries using new terms to alleviate term-mismatch problem. Some studies use *user logs* as an external resource for query expansion [Cui *et al.*, 2002, 2003; Yin *et al.*, 2009]. Whereas, there are some other studies that use WordNet [Collins-Thompson & Callan, 2005; Mandala *et al.*, 1999; Voorhees, 1994]. For example, Nie et al. [Nie & Brisebois, 1996] use WordNet and some user feedback information for document and query expansion. UMLS and MeSH[1], which is a part of UMLS, are also used for document and query expansion [Diem *et al.*, 2007; Gobeill *et al.*, 2009]. We can also find some studies that use the Web (the output of some search engines, or a small snapshot of the Web like WT10g[2] collection) as an external resource for query expansion [Collins-Thompson & Callan, 2005; Diaz & Metzler, 2006; Fang & Zhai, 2006; Yin *et al.*, 2009]. Even Wikipedia[3] is used [Bendersky *et al.*, 2012; Li *et al.*, 2007]. In addition, Qiu et al. [Qiu & Frei, 1993] expand queries based on an automatically-constructed similarity thesaurus. The pseudo-relevance feedback technique can be also seen in this category of external resource based expansion if we suppose that the corpus itself is an external resource.

However, moving from the word-space to the concept-space, or using concepts as indexing terms instead of words, has some side effects on classical IR models, because all retrieval heuristics and statistical studies are well adapted and made depending on words as indexing terms. Accordingly, using concepts instead of words as indexing terms, places us in front of two choices:

[1] www.nlm.nih.gov/mesh/
[2] ir.dcs.gla.ac.uk/test_collections/wt10g.html
[3] en.wikipedia.org

- Proposing an IR model convenient to concepts. This means defining a document and query structures and a matching function compatible with these structures [Baziz, 2005; Le, 2009; Maisonnasse, 2008].
- Still using classical IR models [Ponte & Croft, 1998; Robertson, 1977; Salton *et al.*, 1975], where both documents and queries are bag-of-concepts and the matching function depends, one way or another, on the intersection between documents and queries. This choice requires to study the side effects, which result from using concepts instead of words, on matching functions, and then proposing appropriate solutions.

This section reviews some IR models that are originally proposed to use concepts and to exploit the semantic relations between them (extensive use of knowledge resources). We present three models: [Baziz, 2005] and [Maisonnasse, 2008] that exploit relations at indexing-time, and [Le, 2009] that exploits relations at matching-time. Of course, there are other models that extensively use knowledge resources, e.g. [Roussey, 2001; Styltsvig, 2006].

2.5.1 Graph-Based Matching and Disambiguation

Baziz [Baziz, 2005] proposes a way to represent the semantic content of documents, and to choose, among the candidate concepts, the concept that fits the best the related context or surrounding concepts. The effectiveness of this disambiguation method, namely *DocCore*, is evaluated using a neural network based IR model [Boughanem & Soulé-Dupuy, 1992]. Baziz uses WordNet as a knowledge resource and its synsets as concepts, with exploiting the relations between synsets.

DocCore is a representation of the semantic content of documents. It is a network of synsets. The different phases to construct the DocCore of a document d are:

Candidates extraction. First, longest terms[1] that correspond to at least one entry (synset) in WordNet are identified in text. Each term t has a weight in a document d calculated as follows:

$$weight(t,d) = cf(t,d) \times idf(t)$$

$$cf(t,d) = count(t,d) \times \sum_{st \in sub-terms(t)} \frac{length(st)}{length(t)} \times count(st,d)$$

$$idf(t) = \ln \frac{N}{df(t)}$$

where, N is the number of documents in the collection, $df(t)$ is the number of documents that contain the term t, $count(t,d)$ is the frequency of the term t in the document d, $sub-terms(t)$ is the set of all sub-terms of the term t, and $length(t)$ is the number of words in the term t. Only terms that have a weight greater than 2 are selected to represent the semantic content of documents.

Second, after terms selection, each term normally has multiple senses (polysemy), which means, it corresponds to multiple synsets. For each term, all corresponding synsets are

[1] Baziz defines a term as follows: a term is a non-stop word, which may appear in different grammatical categories (noun, verb, adjective, etc.), or a group of words. A term could correspond to one or several concepts.

used to compose the candidate set for that term. In other words, each term has a set of candidate concepts (synsets) corresponding to the different possible senses of it.

Similarity between concepts. Relations between synsets are not explicitly exploited to build the DocCore. However, they are implicitly used for calculating the semantic similarity between candidates. Several measures are used for the semantic similarity estimation between two candidates, e.g. Resnik measure [Resnik, 1999], Leacock measure [Leacock & Chodorow, 1998], and Lin measure [Lin, 1998].

DocCore construction. As each term corresponds to multiple senses (candidates), multiple networks could be thus constructed. Hence, the question is: Which network does form the best representation of the semantic content of a document? In other words, for each term, which sense is the most appropriate with respect to the context of the document? To select the best candidate for each term, the measure $Cscore$ is used.

$$Cscore(c_i^k) = \sum_{k \neq l, j} \rho(c_i^k, c_j^l)$$

where, c_i^k is a candidate concept of the current term t_k, c_j^l is a candidate concept of another term t_l, and $\rho(c_i^k, c_j^l)$ is the semantic similarity between the two candidates.

Then, for each term the candidate concept that has the greatest $Cscore$ is used for building the DocCore of a document.

To sum up, DocCore is a network of concepts, where for each term, the concept that has the greatest semantic similarity with the concepts of the other terms, is used. Concepts in DocCore are connected together via weighted links. The weights are the value of the semantic similarity between concepts.

Instead of representing documents as networks (DocCore), Baziz also proposes representing them as trees (DocTree). The first phase in obtaining DocTree of a document d is converting that document to a set of concepts (one concept for each term). This conversion is achieved through the following three steps as in DocCore: terms extraction, candidate concepts identification, and concepts disambiguation.

The second phase is to obtain, from WordNet, the minimum sub-tree H_d that contains all concepts of the document d. Here, the relations of WordNet are explicitly used and only the *ISA* relation is used. The previous two phases are repeated for queries.

The third phase is to obtain, from WordNet, the minimum sub-tree H_E that contains the two sub-trees H_d and H_q. In this case, documents and queries are represented by the same tree H_E, but with different node weights. Therefore, two trees H_d^* and H_q^* are obtained, where they contain the same nodes, but with different nodes' weights.

The matching value between a document d and a query q is calculated using the two trees H_d^* and H_q^* and according to the following equation:

$$RSV(d, q) = \sum_{n \in H_E} \mu_{H_q^*}(n) \rightarrow \mu_{H_d^*}(n)$$

where, $\mu_{H_d^*}(n)$ is the weight of the node n (concept) in the document d, $\mu_{H_q^*}(n)$ is the weight of the node n in the query q, and $\rightarrow$ is a fuzzy implication. Many formulas could be used to realize the fuzzy implication, e.g. Dienes implication, Gödel implication, etc.

The experiments of Baziz show that IR models that only use concepts do not perform as well as classical word-based models. In order to obtain better retrieval performance with respect to classical IR models, Baziz needed to either, mix concepts and words, or expand documents and queries using some other concepts extracted from WordNet.

2.5.2 Graph-Based Language Models

Maisonnasse [Maisonnasse, 2008] proposes to use semantic relations, besides concepts, for representing the semantic content of documents and queries. Concretely, Maisonnasse maps the medical text of documents and queries to UMLS concepts, using one of three tools: MetaMap, TreeTagger, or MiniPar[1]. In addition, he uses the semantic relations of UMLS Semantic Network for connecting concepts together.

Maisonnasse proposes two ways to build a concept-based IR model. The first one is to represent a document as a graph and a query as a graph, where nodes are UMLS concepts and edges are the Semantic Relations between the Semantic Types of UMLS Semantic Network. The matching between theses two graphs is a kind of Conceptual Graph projection. The second one is to represent a document as a set of graphs, one for each sentence, and a query as one graph, and then the matching is a graph-based language model.

As an intermediate step, each sentence in text is mapped to concepts, and then these concepts are linked together by semantic relations to construct a graph that represents the semantic content of that sentence. In other words, the intermediate step is converting each sentence in text to a graph. Based on this intermediate representation, two IR models are proposed: *Local Model* and *Global Model*.

Local Model

In this model, graphs that represent all sentences of a document d are concatenated together to form the graph of that document G_d. Each node (concept) c_i in G_d has two weights:

$$P_{ir}(c_i, d) = tf.idf(c_i, d)$$

$$P_{confidence}(c_i, d) = \sum_{ph \in d} P_{confidence}(c_i, ph)$$

where, $P_{ir}(c_i, d)$ is the importance of the concept c_i in the document d. It is calculated using the traditional formula of $tf.idf$ or a variant of it. $P_{confidence}(c_i, ph)$ is the confidence in the process of mapping the sentence ph, or a part of it, to the concept c_i. ph is a sentence in the document d. Each link (semantic relation) r_i in G_d has also two weights:

$$P_{ir}(r_i, d) = tf.idf(r_i, d)$$

$$P_{confidence}(r_i, d) = \sum_{ph \in d} P_{confidence}(r_i, ph)$$

[1]Maisonnasse et al. [Maisonnasse *et al.*, 2009] explore the advantages of merging the output of the three tools: MetaMap, TreeTagger, and MiniPar.

In the same way, the graph of query G_q is constructed. The two graphs G_d and G_q are considered equivalent to two conceptual graphs. Therefore, a projection operation is used for matching. The weights of nodes and links are used for giving a score to the projection operation π, and then for ranking. The matching score between a document d and a query q is:

$$RSV(d,q) = \max_{\pi(q,d)}(\delta(\pi(q,d)))$$

where, $\pi(q,d)$ is a possible projection between the two graphs G_d and G_q, $\delta(\pi(q,d))$ is the degree of correspondence between d and q according to the projection π:

$$\delta(\pi(q,d)) = \sum_{c \in G_q} \delta(c, \pi(c)) + \sum_{r \in G_q} \delta(r, \pi(r))$$

where, c is a concept in G_q, r is a link in G_q, $\pi(c)$ is the concept in G_d that corresponds to c according to π, and $\pi(r)$ is the relation in G_d that corresponds to r according to π.

$$\delta(c, \pi(c)) = P_{ir}(c,q) \times P_{confidence}(c,q) \times P_{ir}(\pi(c),d) \times P_{confidence}(\pi(c),d)$$

$$\delta(r, \pi(r)) = P_{ir}(r,q) \times P_{confidence}(r,q) \times P_{ir}(\pi(r),d) \times P_{confidence}(\pi(r),d)$$

Global Model

In this model, graphs that represent all sentences of a document d are used for building the language model of that document M_d^G, whereas the query q is a graph G_q. In other words, a classical language model is used, but instead of representing documents and queries as bag of words, they are represented by bag of graphs. In addition, the matching score between a document d and a query q represents the ability of document's graphs to generate the query's graph.

$$RSV(d,q) = P(G_q|M_d^G) = P(C_q|M_d^G) \times P(R_q|C_q, M_d^G)$$

where, G_q is the graph that represents the query q, M_d^G is the language model of the document d, which is estimated from all graphs that represent document content (a graph for each sentence), C_q are the nodes (concepts) of the graph G_q, and R_q are the links (semantic relations) of the graph G_q. The model supposes that concepts in the query are statistically independent, then:

$$P(C_q|M_d^G) = \prod_{c \in C_q} P(c|M_d^G)$$

The probability that a document d is capable of generating a concept c, namely $P(c|M_d^G)$, is estimated as follows:

$$P(c|M_d^G) = (1 - \lambda_{con}) \times \frac{D(c)}{D(*)} + \lambda_{con} \times \frac{C(c)}{C(*)}$$

where, $D(c)$ is the frequency of the concept c in the document d, $D(*)$ is the sum of frequencies of all concepts in the document d, $C(c)$ is the frequency of the concept c in the collection C, $C(*)$ is the sum of frequencies of all concepts in the collection C, and λ_{con} is a smoothing

parameter. The model also supposes that relations in the query are statistically independent, then:

$$P(R_q|C_q, M_d^G) = \prod_{r \in R_q} P(r|c_1, c_2, M_d^G)$$

where, c_1 and c_2 are the two concepts in the query that are linked by the relation r. The probability that a document d is capable of generating a relation r, namely $P(r|c_1, c_2, M_d^G)$, is estimated as follows:

$$P(r|c_1, c_2, M_d^G) = (1 - \lambda_{rel}) \times \frac{D(r)}{D(c_1, c_2)} + \lambda_{rel} \times \frac{C(r)}{C(c_1, c_2)}$$

where, $D(r)$ is the frequency of the relation r in the document d, $D(c_1, c_2)$ is the frequency of appearing the two concepts c_1 and c_2 in a graph of the same sentence in the document d, $C(r)$ is the frequency of the relation r in the collection C, $C(c_1, c_2)$ is the frequency of appearing the two concepts c_1 and c_2 in a graph of the same sentence in the collection C, and λ_{rel} is a smoothing parameter.

Concerning the experimental results, Maisonnasse shows that the Local Model does not perform better than classical IR models. In addition, the Global Model performs slightly better than concept based language models, or in other words, using semantic relations does not lead to a huge gain in performance. Semantic relations should be very precise in order to slightly improve the retrieval performance.

2.5.3 Bayesian Network Based Models

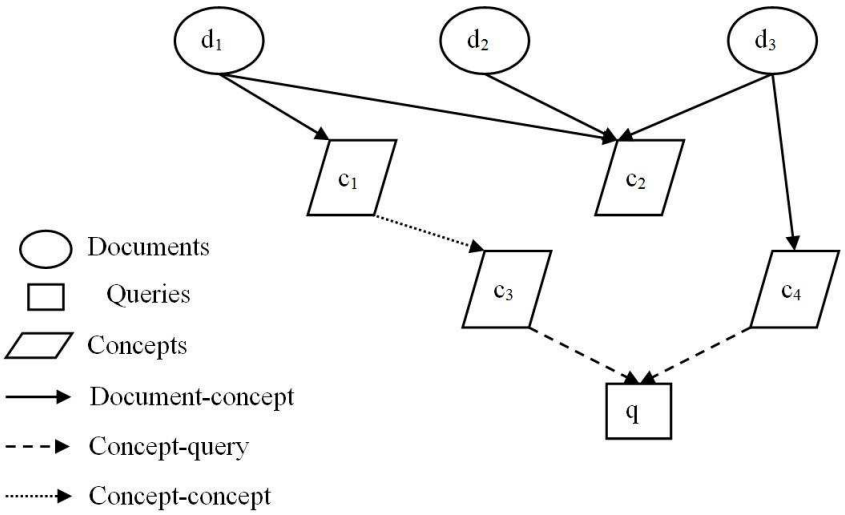
Diem Le [Le, 2009] maps documents and queries to UMLS concepts, and then she places documents and queries on a Bayesian Network. She exploits the relations between concepts to link documents' concepts and queries' concepts. The matching is the actual inference mechanism of Bayesian Networks. According to Diem Le, documents and queries are sets of concepts. Relations are not used till matching-time. Actually, this model only links concepts from documents with concepts from queries. The concepts of documents are not linked together and the same for the concepts of queries. Concretely, MetaMap is used for mapping text to UMLS Meta-thesaurus concepts, and two relations of UMLS Meta-thesaurus are used: *ISA*, *Part-Of*.

Documents, queries, documents' concepts, and queries' concepts, all are arranged in a network structure. The network consists of (Figure 2.3):

- Three types of nodes: documents, concepts, and a query.
- Three types of weighted links: links between documents and their concepts, links between a query and its concepts, and links between documents' concepts and query's concepts.

The matching score between a document d and a query q is estimated using the inference mechanism of Bayesian Network, through initiating the network by interior probabilities, and then updating the probabilities of the other nodes until the posterior probability of the intended node (the query node here) is obtained. Actually, the matching score is estimated as follows: First, the interior probabilities of the network are initiated by observing a particular document d and setting $P(d) = 1$ and the probabilities of other documents to 0; then, the matching score between d and q, or the belief in q giving d as evidence, is calculated according to the following

Figure 2.3: Bayesian Network of Diem Le



recursive equation:

$$RSV(d,q) = P(q|d) = bel(q) = \frac{\sum_{c_i \in q} w(c_i,q) \times bel(c_i)}{\sum_{c_j \in q} w(c_j,q)}$$

where, $w(c_i,q)$ is the weight of the concept c_i in the query q,

$$w(c_i,q) = \frac{w'(c_i,q)}{\sqrt{\sum_{c_j \in q} w'^2(c_j,q)}}$$

$w'(c_i,q)$ is the $tf.idf$ weight of the concept c_i in q, $bel(c_i)$ is the belief in c_i giving the observation of the document d as evidence. Here, three cases can be distinguished:

- If $c_i \in d$ then, $bel(c_i) = w(c_i,d)$.
- If $c_i \in q$ and $\exists c_j \in d$, where (c_i,c_j) are linked by a relation, then,

  $$bel(c_i) = bel(c_h) \times Sim(c_h,c_i)$$

  where, if there are more than one link between c_i and d's concepts then c_h is the concept that has the greatest similarity with c_i.
- Otherwise, $bel(c_i) = 0$.

$w(c_i,d)$ is the weight of the concept c_i in the document d, and $Sim(c_h,c_i)$ is the estimated semantic similarity between c_h and c_i. The Leacock measure [Leacock & Chodorow, 1998] is used for the estimation.

The experimental results in [Le, 2009] show that using concepts instead of words could slightly improve the retrieval performance. In addition, exploiting relations between concepts does not show a considerable gain in performance. However, the experiments in [Le, 2009] were restricted to two very similar test collections, and hence, conclusions are not very solid.

2.5.4 Discussion

The three concept-based IR models, which are presented in the previous subsections, use concepts and exploit relations in order to build a more meaningful representation and matching. Baziz [Baziz, 2005] mainly aims to achieve a context-based conceptual disambiguation. Whereas, Maisonnasse [Maisonnasse, 2008] and Diem Le [Le, 2009] propose two concept-based IR models. Maisonnasse proposes a variation of language models, and Diem Le proposes a Bayesian Network based model. Though interesting and promising, these models are complex, due to the underlying representation used for documents, queries, and matching.

Main conclusions in concept-based IR models are that using concepts alone, to represent the content of documents and queries, is not sufficient. In addition, exploiting relations does not clearly improve the retrieval performance.

2.6 Conclusion

We tried to show in this chapter the importance of knowledge resources in IR, and how they are concretely used and integrated into IR models. The main advantage of using knowledge resources in IR is the ability of using concepts, and exploiting relations between them, in order to describe the content of documents and queries in a more accurate way. In addition, integrating knowledge resources into IR models allows to exploit the information stored in knowledge resources to make relevance judgment more precise and closer to the human way of judgment.

We presented two different techniques to map text to concepts. Furthermore, two knowledge resources WordNet and UMLS are briefly described.

In this chapter, we also presented three IR models that use concepts as indexing terms. Moreover, two of these three models exploit relations between concepts at indexing-time, and the third model exploits relations at matching-time.

However, using concepts and relations in IR has some drawbacks. In general, by using knowledge resources, two more external factors could affect the effectiveness of IR models:

- The precision and the correctness of the text-to-concepts mapping tools: on the one hand, most of these tools are based on NLP techniques to detect noun phrases in text. Noun phrases detection is not a perfect process. On the other hand, the mapping process is an ambiguous process, because the same noun phrase could be mapped to more than one concept. Therefore, we need an extra step to select, among the candidate concepts, the most convenient concept with respect to a specific context.
- The issue of knowledge resources incompleteness: in general, knowledge resources are incomplete, because it is very hard to build a knowledge resource containing all information about a specific domain. As an example, we can see the situation of UMLS. Although UMLS is the largest available resource in the medical domain, several studies show that many concepts and relations are missing in UMLS [Bodenreider *et al.*, 1998, 2001], and there are proposals to compensate this incompleteness. For example, Bodenreider et al. [Bodenreider *et al.*, 2001] postulates that terms with adjectival modifiers are potential hyponyms. They proposes removing the modifiers from a term t_1 to get another term t_2 in a relation of type hyponym with t_1 (t_1 is hyponym of t_2).

To sum up, this chapter shows how a knowledge resource could be exploited and integrated into IR process (either for indexing, matching, or both). This chapter serves as a guide to instantiate our theoretical model in (Chapter 6–P.101).

Chapter 3

Logic-Based IR Models

3.1 Introduction

Formal logics are supposed to be a useful and a valuable mathematical tool in Information Retrieval (IR), because, on the one hand, formal logics are well adapted for knowledge representation [Baader *et al.*, 2003; Barwise, 1989; Barwise & Perry, 1983], and then for building IR models being capable of formally integrating knowledge resources into the retrieval process [Meghini *et al.*, 1993; Nie & Brisebois, 1996]. On the other hand, formal logics are powerful tools for simulating and modeling the inferential nature of the retrieval process [van Rijsbergen, 1986].

Logic-based IR is the formalism that puts all IR notions (document, query, retrieval decision) in a logical framework $\mathcal{L}$. Whatever the choice of logic, even non-classical, most logic-based IR models represent documents and queries as logical sentences, and the retrieval decision as an inference [van Rijsbergen, 1986]. Assume a logic $\mathcal{L}$, if a document d and a query q are logical sentences in $\mathcal{L}$, then d is relevant to q *iff* q is inferable based on d, or in other words, from d, q can be deduced, denoted $d \vdash_{\mathcal{L}} q$, where $\vdash_{\mathcal{L}}$ refers to an inference mechanism related to the logic $\mathcal{L}$.

On the one hand, most formal logics, all classical logics and some non-classical, have only non-fuzzy inference mechanisms (a sentence is inferable or not based on other sentences or axioms). On the other hand, IR is intrinsically an *uncertain* process. Therefore, the non-fuzzy logical inference is quite limited, and a more flexible notion of inference is needed. To overcome these shortcomings, logic-based IR models define, sometimes ad-hoc, another level on top of the logic to represent an uncertainty measure, and consequently, to be able to extend the inference mechanism from binary to continual one.

The first paper that concretely formalizes this field of IR is [van Rijsbergen, 1986]. According to van Rijsbergen, any document d is a set of logical sentences and the query q is a logical sentence. Concerning the retrieval decision, van Rijsbergen argues that the material implication ($\supset$) is problematic and it is not suitable for IR, and he claims that we need a non-classical implication. He uses an example to explain that the probability $P(d \supset q)$ is not the appropriate one to compute the degree to which a document d is relevant, from a system point of view, to a query q. He intuitively postulates that the appropriate measure is the conditional probability $P(q|d)$. Therefore, he expresses the retrieval decision as a condition: 'd *is relevant to* q *iff*

the following condition holds. ***If*** d ***is true then*** q*, denoted* $d \rightarrow q$*, and the uncertainty of it* $U(d \rightarrow q)$ *must be* $P(q|d)$'. The expression 'd is true' means that there should be a formal interpretation somewhere to know that if d is true or not in that interpretation. Actually, instead of dealing with the implication $d \rightarrow q$ and explaining what it exactly refers to, van Rijsbergen directly goes beyond $d \rightarrow q$ and studies how to estimate the uncertainty $U(d \rightarrow q)$. Therefore, we can not see the mapping between the implication $d \rightarrow q$ and the inference $d \vdash q$. However, van Rijsbergen depends on the formal interpretation of the underlying logic to estimate the uncertainty $U(d \rightarrow q)$. Van Rijsbergen uses this idea to evaluate $U(d \rightarrow q)$ and he proposes the well-known *Logical Uncertainty Principle (LUP)*:

> "*Given any two sentences* x *and* y*; a measure of the uncertainty of* $y \rightarrow x$ *relative to a given date set is determined by the minimal extent to which we have to add information to the data set, to establish the truth of* $y \rightarrow x$."

Most studies that came after van Rijsbergen's study [van Rijsbergen, 1986], accept that there is a need to a non-classical logic because the material implication is not suitable for IR. They thus propose some logical models that are complex and hard to implement and test. These models also try, one way or another, to implement the LUP, each model by its way and according the logic that is used.

This chapter is organized as follows: we start, in section 2, by a brief introduction to formal logics and their inference mechanisms. In section 3, we explore the different stands of the implication $d \rightarrow q$, or in other words, the different possible meanings of $d \rightarrow q$. In section 4, we present a panoramic view of the current state of the art of logic-based IR models. We regroup models, first, according to the used concrete logic, and second, according to the mathematical theory that is used to compute the uncertainty $U(d \rightarrow q)$. We conclude this section by a table presenting the definitions of the main IR notions in each model. In section 5, we remember two important IR notions: Exhaustivity and Specificity. Since there is a potential relation between some logics and lattice theory, and since, in our contribution, we exploit lattices to re-express the implication $d \rightarrow q$ and to estimate the uncertainty $U(d \rightarrow q)$, we thus, in section 6, review some lattice-based IR models. We conclude in section 7.

3.2 Introduction to Formal Logics

Formal logics are formal systems consisting of a set of axioms and a set of inference rules e.g. Modus-Ponens. A formal logic $\mathcal{L}$ is normally defined by a formal language and possibly a formal semantic.

The formal language of $\mathcal{L}$ determines the set of all well-formed sentences that can be formed based on a set of atomic elements, namely an alphabet Ω, and a set of connectives Υ, e.g. conjunction $\wedge$, disjunction $\vee$, etc.

Upon this formal language, the inference mechanism $\vdash_{\mathcal{L}}$ related to $\mathcal{L}$ is defined. For any two well-formed sentences s_1 and s_2, $s_1 \vdash_{\mathcal{L}} s_2$ means that based on the axioms of $\mathcal{L}$ and s_1, s_2 can be inferred. The symbol $\vdash_{\mathcal{L}}$ does not belongs to the formal language of $\mathcal{L}$.

The formal semantic of $\mathcal{L}$, if available, is used to give a meaning to the logical sentences and of course to the components of the formal language of $\mathcal{L}$. Assume $\mathcal{L}$ is a classical logic, which is the case in this thesis, the formal semantic is defined based on a set of formal interpretations

Δ. For any sentence s and any interpretation $\delta \in \Delta$, s is either true in δ, denoted $\{\delta\} \models_{\mathcal{L}} s$, or not denoted $\{\delta\} \not\models_{\mathcal{L}} s$. Determining if a sentence is true or not in an interpretation depends on the formal semantic that is given to the alphabet Ω and the connectives Υ. The subset of interpretations $M(s) \subseteq \Delta$ that validate s is called the set of models of s, denoted $M(s) \models_{\mathcal{L}} s$. The formula $M(s_1) \models_{\mathcal{L}} s_2$, or simply, $s_1 \models_{\mathcal{L}} s_2$, means that each model of s_1 is also a model of s_2, or equivalently, in any interpretation if s_1 is true then s_2 is also true. Note also that $\models_{\mathcal{L}}$ does not belong to the formal language of $\mathcal{L}$.

The two symbols $\vdash_{\mathcal{L}}$ and $\models_{\mathcal{L}}$ are related to two different levels of the logic $\mathcal{L}$. While $\vdash_{\mathcal{L}}$ is syntax-related or proof-theoretic notion, the $\models_{\mathcal{L}}$ is semantic-related or model-theoretic notion. However, if the logic $\mathcal{L}$ is *sound* and *complete*, which is the case in all classical logics, then:

- $\mathcal{L}$ is sound: if $s_1 \vdash_{\mathcal{L}} s_2$ then $s_1 \models_{\mathcal{L}} s_2$
- $\mathcal{L}$ is complete: if $s_1 \models_{\mathcal{L}} s_2$ then $s_1 \vdash_{\mathcal{L}}$

Soundness means what is true in the syntax level is also true in the semantic level, and completeness means what is true in the semantic level is also true in the syntax level.

Moreover, in classical logics the formula $s_1 \vdash_{\mathcal{L}} s_2$ is equivalent to $\models_{\mathcal{L}} s_1 \supset s_2$, where $\supset$ is the material implication. Note that the two symbols $\vdash_{\mathcal{L}}$ and $\models_{\mathcal{L}}$ do not belong to the formal language of $\mathcal{L}$.

3.3 Inference Status

Sebastiani [Sebastiani, 1998] reviews the different stands related to the IR implication $d \rightarrow q$. In other words, what does the operation '$\rightarrow$' refer to? Suppose $\mathcal{L}$ is a logical framework. A document d and a query q are logical sentences in $\mathcal{L}$. Accordingly, the logical status of the inference $d \rightarrow q$ can refer to one of the following stands:

Truth. $d \rightarrow q$ is true in a particular interpretation $\{\delta\}$ of $\mathcal{L}$, denoted $\{\delta\} \models_{\mathcal{L}} d \rightarrow q$. In IR, truth means that d is relevant to q *iff* there is at least one interpretation in which d and q are true. In this case, the symbol '$\rightarrow$' must belong to the formal language of $\mathcal{L}$.

Logical consequence. $d \rightarrow q$ refers to that q is a logical consequence of d in $\mathcal{L}$, denoted $M(d) \models_{\mathcal{L}} q$ or simply $d \models_{\mathcal{L}} q$, where $M(d)$ is the set of interpretations in which d is true. In IR, logical consequence means that d is relevant to q *iff* all interpretations that validate d also validate q. In this case, the symbol '$\rightarrow$' does not belong to the formal language of $\mathcal{L}$.

Validity. $d \rightarrow q$ is valid in $\mathcal{L}$, denoted $\models_{\mathcal{L}} d \rightarrow q$, means that $d \rightarrow q$ is true in any interpretation of $\mathcal{L}$. In IR, d is relevant to q *iff* the logical sentence $d \rightarrow q$ is always true. In this case, the symbol '$\rightarrow$' must belong to the formal language of $\mathcal{L}$.

Derivability based on a knowledge. $d \rightarrow q$ is derivable by applying the inference rules of $\mathcal{L}$ to the axioms of $\mathcal{L}$ and to the set of sentences Γ, denoted $\Gamma \vdash_{\mathcal{L}} d \rightarrow q$. This status is also similar to the derivability of q from d and Γ, denoted $\Gamma \cup \{d\} \vdash_{\mathcal{L}} q$. In IR, d is relevant to q *iff* the logical sentence $d \rightarrow q$ is obtainable through applying the inference rules of $\mathcal{L}$ to its axioms and to an external knowledge represented by the set of sentences Γ. In this case, the symbol '$\rightarrow$' must belong to the formal language of $\mathcal{L}$.

Derivability. $d \rightarrow q$ refers to that q is derivable from d, denoted $d \vdash_{\mathcal{L}} q$. In other words, q can be obtained by applying the inference rules of $\mathcal{L}$ to its axioms and to d. In IR, d is relevant to q *iff* applying the inference rules of $\mathcal{L}$ to its axioms and to d is sufficient to obtain q. The main difference between this status and the previous one, is the need or not to an external knowledge to obtain q from d. In this case, the symbol '$\rightarrow$' does not belong to the formal language of $\mathcal{L}$.

Theorem. $d \rightarrow q$ is a theorem in $\mathcal{L}$, denoted $\vdash_{\mathcal{L}} d \rightarrow q$, or in other words, $d \rightarrow q$ can be obtained by applying the inference rules of $\mathcal{L}$ to its axioms only. In IR, d is relevant to q *iff* applying the inference rules of $\mathcal{L}$ to its axioms only is sufficient to obtain $d \rightarrow q$. In this case, the symbol '$\rightarrow$' must belong to the formal language of $\mathcal{L}$.

Truth, logical consequence, and validity are *model-theoretic* notions, i.e. related to the formal semantic and interpretation of $\mathcal{L}$, whereas, derivability and theorem are *proof-theoretic* notions, i.e. related to the formal language of $\mathcal{L}$. In any logic $\mathcal{L}$ (*sound* and *complete*), derivability is equivalent to logical consequence, and validity to theorem. Moreover, in classical logics, the logical consequence $d \models q$ and the validity $\models d \supset q$ are equivalent, where '$\supset$' is the material implication. Accordingly, $d \vdash q$ is equivalent to $\vdash d \supset q$.

Note that in validity and theorem the connective '$\rightarrow$' is a part of the formal language $\mathcal{L}$, whereas, in derivability and logical consequence '$\rightarrow$' is replaced by either '$\vdash$' or '$\models$', which are meta-symbols and they are not a part of the formal language.

Sebastiani also argues that validity and logical consequence are more suitable than truth for IR. More precisely, while validity and logical consequence are *form-based*, truth is a *content-based* notion and consequently less efficient to check. For example, a logical sentence like $s \vee \neg s$ is valid whatever s refers to, and hence validity can be formally checked, i.e. validity is a form-based notion. However, the truth of an arbitrary logical sentence s can be checked only with respect to a particular interpretation. An interpretation normally represents a description of the world. Therefore, if we assume that we have a total knowledge about the world, which is a very strong assumption, then the truth of s can be checked, otherwise, if we have a partial knowledge about the world, which is the general case, then it is difficult to check the truth of s in a particular interpretation. In IR, the total knowledge refers to the close world assumption where there is only one unique interpretation in which d is true, whereas, partial knowledge refers to the open world assumption where there are many interpretations in which d is true, in other words, for terms that do not appear in d, there is not enough knowledge to decide if d is about these terms or not (see Appendix C–P.185).

3.4 Overview of Logic-Based IR Models

The IR literature contains many studies that aim to concretize the LUP of van Rijsbergen [van Rijsbergen, 1986]. They use different types of logic, classical and non-classical. The classical logics used range from Propositional Logic ($\mathcal{PL}$) to First-Order Logic ($\mathcal{FL}$), passing through Description Logic ($\mathcal{DL}$). Actually, it is difficult to say that a particular logic is better than others for IR. Furthermore, there is a trade-off between the expressive power of a logic and the complexity of its deduction algorithms, where the more expressive the formal language of a logic is, the more complex or costly its deduction process is.

For any concrete logic-based IR model, and whatever the used logic is, the model must clearly and precisely define the following four main components: document, query, retrieval decision $d \rightarrow q$, and the way of computing the uncertainty measure $U(d \rightarrow q)$. In this section, we show how each model represents the previous four IR components.

Normally, there are two axes to present logical IR models: the formal logic that is concretely used, and the mathematical theory that is used to compute the uncertainty. In this section, we regroup models according to the type of logic that is used, e.g. propositional logic, description logic, etc. In each model, we show the exact definition of: d, q, $d \rightarrow q$, and $U(d \rightarrow q)$.

3.4.1 Models based on Propositional Logic

Losada et al. [Losada & Barreiro, 2001] use Propositional Logic ($\mathcal{PL}$)[1] to build an IR model. Each indexing term is an atomic proposition, and it could be either true or false in a particular document or query. A document d is a logical sentence formed using the indexing terms. A query q is also a logical sentence. The retrieval decision is a logical consequence or entailment, where d is relevant to q *iff* $d \models q$. As the model is proposed in $\mathcal{PL}$ framework then $d \models q$ is equivalent to $\models d \supset q$. In the formal semantic of $\mathcal{PL}$, $d \models q$ means that every model of d is also a model of q, or in other words, for every interpretation δ in which d is true, q is also true.

To estimate the uncertainty of $d \models q$, denoted $U(d \models q)$, Losada et al. use Belief Revision (BR), which is a technique to formally express the notion of proximity between logical sentences. In other words, BR deals with updating an existing knowledge K with a new piece of information s, denoted $K \circ s$, where, if there is no contradiction between K and s then the updated knowledge becomes $K \circ s = K \wedge s$, otherwise, BR deals with the *minimal change* that should be made on K in order to build updated knowledge K' that does not contradict with s, $K \circ s = K' \wedge s$. This last notion of *minimal change* is a central notion for IR.

There are many BR techniques that deal with the syntax of the logical language, namely formula-based approaches, and other techniques dealing with the formal semantic of the logic, namely model-based approaches. More precisely, Losada et al. use Dalal's BR operator, denoted $\circ_D$, which is one of the model-based approaches of BR. According to this operator, giving two interpretations δ_i and δ_j, the distance between them, denoted $dist(\delta_i, \delta_j)$, is the number of atomic propositions in which the two interpretations differ. For example, assume that our alphabet Ω contains three atomic propositions $\Omega = \{a, b, c\}$. Assume we have two interpretations: $\delta_i = \{a, b\}$ which means that a and b are true whereas c is implicitly false, $\delta_j = \{a, c\}$ which means that a and c are true whereas b is implicitly false. The distance between δ_i and δ_j is:

$$dist(\delta_i, \delta_j) = |(\delta_i \cup \delta_j) \setminus (\delta_i \cap \delta_j)|$$

Hence, the distance between a logical sentence s and an interpretation δ is calculated as follows:

$$Dist(M(s), \delta) = \min_{m \in M(s)} dist(m, \delta)$$

$M(s)$ is the set of models of s, or equivalently, the set of interpretations in which s is true.

To estimate the uncertainty of the retrieval decision $U(d \models q)$ using Dalal's BR operator, Losada et al. distinguish between two modes of revision:

[1] See (Appendix C–P.185) for more information about the formal language of $\mathcal{PL}$ and its formal semantic.

- Revising q by d, denoted $q \circ_D d$. Here, there are two cases:
  - q has several models $M(q)$ whereas d has only one unique model m_d (Close World Assumption).
    $$distance(d, q) = Dist(M(q), m_d)$$
  - q has several models $M(q)$ and d also has several models $M(d)$ (Open World Assumption).
    $$distance(d, q) = \frac{\sum_{m \in M(d)} Dist(M(q), m)}{|M(d)|}$$

  Now the similarity between d and q is:
  $$BRsim(d, q) = 1 - \frac{distance(d, q)}{k}$$
  where k is the number of atomic propositions appearing in q.
- Revising d by q, denoted $d \circ_D q$.
  $$distance(q, d) = \frac{\sum_{m \in M(q)} Dist(M(d), m)}{|M(q)|}$$
  Now the similarity between d and q is:
  $$BRsp(d, q) = 1 - \frac{distance(q, d)}{l}$$
  where l is the number of atomic propositions appearing in d.

According to Losada et al., the final retrieval score is the weighted sum of the two similarities:

$$U(d \models q) = \alpha \times BRsim(d, q) + (1 - \alpha) \times BRsp(d, q)$$

The main problem in the previous equation is that computing $BRsim(d, q)$ and $BRsp(d, q)$ is exponential and time consuming, because the number of models $M(d)$ and $M(q)$ are exponential in the number of atomic propositions or equivalently indexing terms.

Losada et al. propose to rewrite the logical sentences in Disjunctive Normal Form (DNF), and instead of computing the distance between d and q based on interpretations, they compute it based on clauses. A sentence s in DNF is a disjunction of clauses $c_1 \vee \cdots \vee c_n$ and each clause c_i is a conjunction of literals $l_1 \wedge \ldots l_m$ and each literal l_j is either an atomic proposition a_j or its negation $\neg a_j$. The distance between two clauses is the number of atomic propositions that are positive in one clause and negative in the other, and it is computed as follows:

$$CDist(c_i, c_j) = |\{l \in c_i | \neg l \in c_j\}|$$

This new approach is efficient to compute, and then applying the model to document collections of large scale is feasible. However, this approach is only applicable in the case where each of d and q is only one clause. In the general case where d and q are DNF sentences, Losada et al. illustrate that the computation of query models is still needed, and the algorithm is still

exponential. To overcome this problem, they propose a simplification, but in this case the technique of computing the uncertainty U is no more BR, it becomes ad-hoc.

To sum up, since atomic propositions correspond to the indexing terms, the distance measure calculates how many different terms exist between d and q. Losada et al. claim that their model is equivalent to the Vector Space Model (VSM) [Salton *et al.*, 1975] with a more powerful language to represent the content of documents and queries, but without ability to represent term weights.

Losada et al. [Losada & Barreiro, 2003] apply their model to a large collection of TREC[1] (about 170000 documents). They build the DNF sentences of queries as follows: each part of the query (i.e. title, description, narrative) corresponds to one clause and q is $title \vee description \vee narrative$. DNF logical sentences are also built for documents. The retrieval performance of their model is better than the classical *tf.idf* based VSM. However, they do not compare their model with other IR models, and they do not even try with other document collections. They also claim that the most of gain in performance comes from building a rather complex document and query representation.

3.4.2 Models based on Probabilistic Argumentation Systems

Probabilistic Argumentation Systems (PAS) augments Propositional Logic ($\mathcal{PL}$) by a mechanism to capture uncertainty using probability. The uncertainty is captured through using special propositions called *assumptions*. Uncertain rules in PAS are generally defined as follows:

$$a \wedge s_1 \rightarrow s_2$$

where '$\rightarrow$' is the material implication, s_1 and s_2 are logical sentences formed based on a set of propositions (the alphabet of propositional logic of PAS), and a is a logical sentence formed based on another set of propositions (the assumptions). The two sets of propositions (alphabet and assumptions) are disjoint. The above rule means that s_1 implies s_2 under some circumstances represented by a. It is also possible to link probability to assumptions, where $P(a)$ means $P(s_2|s_1)$ and not $P(s1 \rightarrow s_2)$.

A knowledge base K is a set of uncertain rules of the previous form. It is possible to check if a knowledge base K *supports* or *discounts* a particular hypothesis h, where h is any logical sentence. The support of a hypothesis h in a knowledge base K, denoted $sp(h, K)$, is the disjunction of the assumptions α_i, where for any assumption α_i we have $(\alpha_i \wedge K) \models h$. The *degree of support* of a hypothesis h in a knowledge base K, denoted $dsp(h, K)$, refers to the degree to which K supports h knowing that h does not contradict with K. The degree of support $dsp(h, K)$ is the quantitative or numerical expression of the support $sp(h, K)$, and $dsp(h, K)$ is a type of conditional probability.

Picard et al. [Picard & Savoy, 2000] use PAS to build an IR model. According to Picard et al., documents, queries, and the indexing terms are atomic propositions. The content of documents and queries is represented by a set of rules of the following form:

$$a_{ij} \wedge d_i \rightarrow t_j$$

[1] Text REtrieval Conference (http://trec.nist.gov/).

which means that the document d_i is about or indexed by the term t_j with a degree of uncertainty $P(a_{ij}) = P(t_j|d_i)$. Semantic relations between terms are represented using the following rule:

$$b_{ij} \wedge t_i \rightarrow t_j$$

which means that there is a relation between t_i and t_j, and $P(b_{ij})$ is the strength of this relation. The strength of a relation is related to the type of this relation.

The set of previous rules form the IR knowledge base K. The retrieval decision $d \rightarrow q$ is the material implication $d \supset q$. The uncertainty $U(d \supset q)$ is estimated in two ways:

- symbolically: $U(d \supset q) = sp(q, K \wedge d)$.
- numerically: $U(d \supset q) = dsp(q, K \wedge d)$.

In both ways, the document d is assumed as a fact or it is observed. The knowledge base K is updated according this observation to become $K \wedge d$. Actually, the model captures the degree to which the new knowledge base $K \wedge d$ supports the query q.

The formalism presented by Picard et al. is capable of representing the inter-terms relationships through rules of the form: $b_{ij} \wedge t_i \rightarrow t_j$. However, Picard et al. put more emphasis on the ability of their formalism to represent the inter-documents relationships, e.g. hyperlinks, citations, etc. They use rules of the form: $l_{ij} \wedge D_i \rightarrow D_j$ to represent this type of relations, where D_i is related to D_j by a link directed from D_i to D_j, and the probability $P(l_{ij})$ reflects the type of the relation and its strength. The main disadvantage is that estimating $P(a_{ij})$, $P(b_{ij})$, $P(l_{ij})$ in the previous rules requires the availability of relevance information, and the probability $P(a_{ij})$ in the rule $a_{ij} \wedge d_i \rightarrow t_j$ is finally estimated using a variant of classical *tf.idf* measure. In general, the main difficulty in this study is the probability estimation.

Experimentally, it is not possible to draw clear conclusions from the study of Picard et al., because they only apply their model to the corpus CACM, which is a very small collection of documents (only 3204 documents).

3.4.3 Models based on Propositional Modal Logic

Modal logics define two modalities *necessary* and *possible* that could be added to any logic. There are thus Propositional Modal Logic ($\mathcal{PML}$), First-order Modal Logic, etc. We here focus on $\mathcal{PML}$. Modal logics use the *Possible Worlds* (PW) semantic to give a meaning to the previous two modalities [Kripke, 1963]. Worlds are connected through accessibility relations. Before presenting IR models that are built upon PW semantic, let us first give a brief introduction to PW semantic and to the related probabilistic technique, namely *Imaging* [Gardenfors, 1982; Lewis, 1973].

3.4.3.1 Possible Worlds & Imaging

In (Appendix C–P.185), we present the formal interpretation and semantic of Propositional Logic ($\mathcal{PL}$). Kripke's semantic [Kripke, 1963], or PW semantic, is another way to give a formal semantic to logical sentences. It is principally suggested to give a formal semantic to $\mathcal{PML}$.

PW semantic is the structure $\langle W, R \rangle$, where W is a non-empty set of what conventionally called *Possible Worlds*, and R is a binary relation $R \subseteq W \times W$ and it is conventionally called *accessibility relation*.

The structure $\langle W, R, \Vdash \rangle$ defines the Kripke's system, where $\Vdash$ determine for any sentence s if it is true in a world w, denoted $w \Vdash s$, or not, denoted $w \nVdash s$. For classical $\mathcal{PL}$ connectives (conjunction $\wedge$, disjunction $\vee$, negation $\neg$), their semantic in a Kripke's system is defined as follows:

- Negation: $w \Vdash \neg s$ *iff* $w \nVdash s$.
- Conjunction: $w \Vdash s_1 \wedge s_2$ *iff* $w \Vdash s_1$ and $w \Vdash s_2$.
- Disjunction: $w \Vdash s_1 \vee s_2$ *iff* $w \Vdash s_1$ or $w \Vdash s_2$.

As we said, $\mathcal{PML}$ defines two special unary operators, *Necessarily* ($\Box$) and *Possibly* ($\Diamond$), in order to represent two modalities. The formal semantic of the previous two operators in a Kripke's system is:

- Necessarily: $w \Vdash \Box s$ *iff* for any world w' satisfying $(w, w') \in R$, $w' \Vdash s$.
- Possibly: $w \Vdash \Diamond s$ *iff* there exists a world w' satisfying $(w, w') \in R$, $w' \Vdash s$.

The notation $(w, w') \in R$ means that the world w' is accessible from w through the accessibility relation R. The properties of the accessibility relation R, or equivalently, the way of interaction between the added operators ($\Box$,$\Diamond$) and the classical operators ($\wedge$,$\vee$,$\neg$), determines different families of $\mathcal{PML}$, having different expressive powers. Some families of $\mathcal{PML}$ are *sound* and *complete*.

Imaging is a process developed in the framework of modal logics. It enables the evaluation of a conditional sentence $a \rightarrow b$ without explicitly defining the operator '$\rightarrow$'. To define imaging, let us first assume that a probability distribution P is defined on the set of possible worlds W as follows:

$$\sum_{w \in W} P(w) = 1 \tag{3.1}$$

Imaging transfers probabilities from some worlds to other worlds and builds a new probability distribution. In imaging, any logical sentence s can only be true or false in a particular world $w \in W$. Therefore, we define $w(s) = 1$ if s is true in w ($w \Vdash s$), or $w(s) = 0$ otherwise. We also define w_s to be the most similar world to w where s is true, $w_s(s) = 1$. The most similar, the closest, etc., they are notions related to the definition of the accessibility relation R. *Generalized Imaging* [Gardenfors, 1982] relaxes the previous two assumptions, where:

- The truth of s in a world w is no more binary.

$$w(s) = \begin{cases} 0 & \text{if} \quad w \nVdash s \\ > 0 & \text{otherwise} \end{cases}$$

  In addition, for any logical sentence s, the following condition holds, $\sum_{w \in W} w(s) = 1$.
- There could be more than one most similar world. Therefore, w_s is no more one distinct world, it is now a set of worlds, where w_s are the worlds the most similar to w where s is true $\forall w' \in w_s, w'(s) > 0$.

After defining the probability on worlds (Equation 3.1), the probability can also be defined on logical sentences. For any logical sentence s:

$$P(s) = \sum_{w \in W} P(w) \times w(s)$$

Now, the imaging on a logical sentence s is the process of moving probabilities from the worlds where s is false to the most similar worlds where s is true. Imaging on s creates a new probability distribution P_s, which is defined as follows:

$$P_s(w') = \sum_{w \in W} P(w) \times I(w, w')$$

where

$$I(w, w') = \begin{cases} 1 & \text{if} \quad w' = w_s \\ 0 & \text{otherwise} \end{cases}$$

In generalized imaging:

$$P_s(w') = \sum_{w \in W} P(w) \times P^w(w') \times I(w, w')$$

where

$$I(w, w') = \begin{cases} 1 & \text{if} \quad w' \in w_s \\ 0 & \text{otherwise} \end{cases}$$

and $P^w(w')$ is the weight of the link between w and w', or is the portion of the probability $P(w)$ that must be transferred to w', because in generalized imaging w_s is a set of worlds and $P(w)$ must be distributed on all worlds in w_s.

The probability of a condition $s_1 \rightarrow s_2$ is defined as follows [Amati *et al.*, 1992; Crestani, 1998; Crestani & Rijsbergen, 1995]:

$$P(s_1 \rightarrow s_2) = P_{s_1}(s_2) = \sum_{w \in W} P_{s_1}(w) \times w(s_2) \tag{3.2}$$

Some researchers represent the retrieval decision $d \rightarrow q$ as a condition. Therefore, (Equation 3.2) is an important technique to estimate the uncertainty $U(d \rightarrow q)$ in a probabilistic way.

3.4.3.2 Using Possible Worlds Semantic

Nie [Nie, 1988, 1989] uses Propositional Modal Logic ($\mathcal{PML}$), or Kripke's formal semantic [Kripke, 1963], to refine the Logical Uncertainty Principle (LUP) that is proposed by van Rijsbergen [van Rijsbergen, 1986]. Nie defines his logic-based IR model within the formal semantic layer. The inference mechanism in Nie's model has no direct correspondence in the formal language of $\mathcal{PML}$.

Nie assumes that a document d is a set of logical sentences, or equivalently, it corresponds to a possible world. A query q is a set of propositions or a logical sentence. Any proposition a is true in d if it appears in d.

To evaluate $U(d \rightarrow q)$, the model starts from the initial world d (or d_0), if q is not satisfied in d_0, then using accessibility relations the model goes from d_0 to d_1. If q is still not satisfied in d_1 the model goes from d_1 to d_2, and so on, until the model arrives to d_n that satisfies q. Actually, there are many paths from d_0 to d_n. Therefore, to calculate the certainty of the implication $U(d \rightarrow q)$, the path of the minimal distance is chosen. A general measure to calculate the distance from d_0 to d_n, denoted $dis(d_0, d_n)$, is defined as a function of the elementary distances $dis(d_i, d_{i+1})$.

Contrariwise, instead of considering documents as possible worlds, it is possible to consider queries; in this case, the model must find the path from q_0 to q_n that satisfies d. In both cases, the accessibility relations between possible worlds could be linguistic relations. For example, the model could transfer from q_i to q_{i+1} by adding the synonymous terms of the indexing terms of q_i to q_{i+1} (query expansion). Nie reformulated the LUP as follows:

> "*Given any two information sets x and y; a measurement of the uncertainty of $y \rightarrow x$ relative to a given knowledge set K, is determined by the minimal extent E to which we have to add information to y, to establish the truth of $(y + E) \rightarrow x$.*"

The implication $d \rightarrow q$ is thus equivalent to find a path from the initial possible world d to another possible world d_n that satisfies q, whereas, the uncertainty $U(d \rightarrow q)$ is equivalent to the cost, or the total distance, of this path.

3.4.3.3 Using Imaging

In the studies of Nie [Nie, 1988, 1989], the distance between worlds, or the cost of accessibility relations is arbitrary defined. However, Nie [Nie, 1992] defines two sources of uncertainty in a more formal way[1]:

- The truth of a proposition a in a world w is not binary, $w(a) \in [0, 1]$. The function $w()$ is recursively defined on any logical sentence, and Nie proved that it is a probability.
- The strength of the accessibility relation between two worlds w and w' depends on the type of the semantic relation, e.g. synonymy, generalization, specialization, etc., that causes the transformation from w to w', where $\sum_{w' \in W} P^w(w') = 1$.

Nie [Nie, 1992] uses probability (imaging) to estimate the logical uncertainty $U(d \rightarrow q)$. According to Nie, a document d is a possible world and a query q is a logical sentence. First, the model of Nie makes an imaging on d to transform probabilities from the worlds that have an accessibility relation with d to d. To compute the logical uncertainty $U(d \rightarrow q)$, Nie uses (Equation 3.2), where:

$$U(d \rightarrow q) = \sum_{w \in W} P_d(w) \times w(q)$$

The accessibility strength $P^w(w')$ is used to compute the truth function $w()$, and it is also used in the imaging process to choose the closest world to d in order to build the new probability distribution P_d.

Unlike Nie who represents a document d as a possible world, a query q as a logical sentence, and d in that case is relevant to q *iff* there is a path from d to d_n that satisfies q [Nie, 1988, 1989], Crestani et al. [Crestani & Rijsbergen, 1995] and Crestani [Crestani, 1998] assume that each indexing term $t \in T$ is a possible world. A document d is true in t if t appears in d. A query q is true in t if t appears in q. Crestani et al. define t_d as the closest term to the term t where d is true.

Crestani et al. use the imaging technique to move probabilities from the terms that do not appear in d to the terms that appear in d. They build a new probability distribution P_d from

[1] For readability reasons, we use the notation proposed in (Section 3.4.3.1) instead of notations used in original papers.

the original distribution P by imaging on d. Crestani et al. prove that the logical uncertainty $U(d \rightarrow q)$ can be estimated as follows:

$$U(d \rightarrow q) = P_d(q) = \sum_{t \in T} P_d(t) \times t(q)$$

where $t(q) = 1$ if t appears in q or $t(q) = 0$ otherwise.

For the prior probabilities of terms, Crestani et al. use the terms discriminative power measures, like IDF:

$$\forall t \in T, P(t) = -IDF(t) = -\log \frac{n_t}{N}$$

where N is the total number of documents and n_t is the number of documents that are true in the possible world t.

The strength of the accessibility relation between two terms t_i and t_j, which computes the similarity between them, is estimated using the Expected Mutual Information Measure (EMIM) [van Rijsbergen, 1977].

Zuccon et al. [Zuccon *et al.*, 2009] show that imaging based IR models, as presented in [Crestani & Rijsbergen, 1995], have a retrieval performance much lower than the performance of some classical IR models.

3.4.3.4 Fuzzy Propositional Modal Logic

Nie et al. [Nie & Brisebois, 1996] define the fuzzy accessibility relations between two possible worlds and the fuzzy truth value of a proposition in a possible world, in order to build an IR model. They represent a document d as a possible world and a query q as a logical sentence. They redefine the two functions $P^w(w')$ and $w(s)$ as follows:

- The function $P^w(w')$ estimates the fuzzy accessibility degree between two worlds w and w', where $P^w(w) = 1$.
- For any logical sentence s, the function $w(s)$ gives the fuzzy truth value of the sentence s in the world w. This function is built based on $C_a(w)$ which represent the fuzzy truth value of an atomic proposition a in a world w.

The retrieval decision is defined in the same way as in [Nie, 1988, 1989, 1992], whereas, the logical uncertainty $U(d \rightarrow q)$ is defined as follows:

$$U(d \rightarrow q) = d(\Diamond^n q)$$

where

$$d(\Diamond^n q) = \sup_{w \in W} \Delta[P^d(w), w(\Diamond^{n-1} q)]$$

Δ is a triangular norm function and

$$d(\Diamond q) = \sup_{w \in W} \Delta[P^d(w), w(q)]$$

To establish the fuzzy degree of accessibility between two possible worlds $P^w(w')$, Nie et al. propose an automatic way to learn the strength of inter-terms relationships, where $P^w(w')$

equals the strength of the relation that causes the transition from w to w'. The main problem is that estimating these weights requires a non-negligible amount of user feedback information.

Experimentally, since Nie et al. apply their model to the corpus CACM, which is a very small collection of documents (only 3204 documents), then it is not possible to draw clear conclusions from their study.

3.4.3.5 Conclusion

Even though using Kripke's semantic and its related imaging technique seems to be an interesting theoretical choice, IR models based on possible worlds and imaging have some disadvantages. These models are totally defined in the formal semantic side, and some operations have no direct correspondence in the formal language of the logic (syntax). For example, in the condition of the form $d \rightarrow q$ the connective '$\rightarrow$' has no correspondence in the language of propositional modal logic. In addition, most models directly define the logical uncertainty $U(d \rightarrow q)$ without defining what the connective '$\rightarrow$' refers to. This point could also be assumed as an advantage because it allows us to exceed the task of defining '$\rightarrow$'.

It is also not easy to define the prior probability distribution on worlds $P(w)$, and what it refers to. Furthermore, defining accessibility relations and their related cost or distance measure is also a heavy task and need a lot of study. Finally, experiments on large document collections show poor retrieval performance of these models.

3.4.4 Models based on Description Logic

Description Logic ($\mathcal{DL}$) is a family of languages to represent knowledge. It is a sort of logic more expressive than Propositional Logic ($\mathcal{PL}$) but it has more efficient reasoning than First-Order Logic ($\mathcal{FL}$). While the reasoning in $\mathcal{FL}$ is NP-complete [Amati & Ounis, 2000; Chein & Mugnier, 1992], there are some variants of $\mathcal{DL}$ having polynomial time complexity [Koller *et al.*, 1997], deterministic polynomial time [Lukasiewicz, 2008], PSpace-complete [Qi & Pan, 2008], or even $O(n \log n)$ in the *ALN* family [Sebastiani & Straccia, 1991]. A logical sentence in $\mathcal{DL}$ is a formulation of building blocks and connectives (operators) according to predefined rules. Many families of $\mathcal{DL}$ can be defined based on the allowed operators. Of course, in any family of $\mathcal{DL}$ there is a trade-off between the expressive power and the efficiency of related algorithms. $\mathcal{DL}$ contains three building blocks:

- *Individuals*, which are concrete objects in the real life, e.g. Alex, Bob, etc.
- *Concepts*, which define classes of objects, e.g. Dogs, Cats, Whit, etc.
- *Roles*, which define the role of objects or classes in relations, e.g. Husband, Author, etc.

Building blocks are linked together through a set of allowed operators. Concerning concepts, there are many operators e.g. *Intersection* ($\sqcap$), *Union* ($\sqcup$), etc. For example, '$Dogs \sqcap White$' defines the white dogs concept. The two quantifiers *Universal* ($\forall$) and *Existential* ($\exists$) are also used to link roles with concepts, e.g. '$\forall Author.Human$' means that the authors of any object are humans. Two types of reasoning are defined in $\mathcal{DL}$:

- Subsumption between two concepts or roles ($\sqsubseteq$), e.g. '$Dogs \sqsubseteq FourLegsAnimals$' means that all dogs are four-legs animals but not the inverse.

- Role or concept assertion (:), which links concepts and roles to their individuals, e.g. '$Alex : Dogs$' means that Alex is a dog.

In (Appendix C–P.185), we present the formal semantic of $\mathcal{PL}$, and in (Section 3.4.3.1–P.46) we present the possible world semantic that is proposed in the framework of modal logic. Here, we present the formal semantic or interpretation that is used to give sense to logical sentences in $\mathcal{DL}$. We define a set of elements Δ^I that will represent the domain of interpretation, and then we define the interpretation function $.^I$, which maps:

- every individual a to an element in the domain Δ^I, where $a^I \in \Delta^I$.
- every concept C to a subset of elements $C^I \subseteq \Delta^I$. Two special concepts exist: the concept that contains all individuals ($\top$), where $\top^I = \Delta^I$, and the empty concept ($\bot$), where $\bot^I = \emptyset$.
- every role R to a subset of domain Cartesian product $R^I \subseteq \Delta^I \times \Delta^I$.

Concerning operators, their formal semantic is defined as follows:

- $(C \sqcup D)^I = C^I \cup D^I$
- $(C \sqcap D)^I = C^I \cap D^I$
- $(\neg C)^I = \Delta^I \setminus C^I$
- $(\forall R.C)^I = \{x \in \Delta^I | \forall y \in \Delta^I, (x,y) \in R^I, y \in C^I\}$
- $(\exists R.C)^I = \{x \in \Delta^I | \exists y \in \Delta^I, (x,y) \in R^I, y \in C^I\}$

We say that an interpretation I is a model of a $\mathcal{DL}$ logical sentence s *iff* s is true in I, denoted $I \models s$. The truth of a logical sentence in an interpretation I is determined according to the following rules:

- $I \models a : C$ *iff* $a^I \in C^I$.
- $I \models (a,b) : R$ *iff* $(a^I, b^I) \in R^I$.
- $I \models C \sqsubseteq D$ *iff* $C^I \subseteq D^I$.

A Knowledge Base (KB) is the pair (T, A), where T is the terminological box that contains the definitions of concepts, roles, and the relations between them e.g. $C \sqsubseteq D$, and A is the assertion box that contains the relations between concepts and roles from one hand and individuals from the other hand, e.g. $a : C$, $(a,b) : R$.

Meghini et al. [Meghini *et al.*, 1993] use a special kind of $\mathcal{DL}$, called MIRTL, to realize the logical implication $d \rightarrow q$, and thus building an IR model. According to them, a document d is represented as an individual, or in other words, a document d is the only instance of a concept D which is the intersection of all concepts where d is asserted to be an instance of, formally $D \doteq \bigsqcap_{d:X_i} X_i$. A query q is represented as a concept. The relevance judgment between a document d and a query q is mapped either to:

- the individual d is an instance of the concept q, denoted $d : q$.
- the concept D that only contains the document d is subsumed by the concept q, denoted $D \sqsubseteq q$.

The two decisions $d : q$ or $D \sqsubseteq q$ are binary decisions, that means, the knowledge base KB either satisfies them or not. Therefore, $\mathcal{DL}$ alone does not support the partial or uncertain decision, which is the adequate type of decision in IR. To calculate the logical uncertainty $U(d \rightarrow q)$, $\mathcal{DL}$ is extended by probability [Lukasiewicz, 2008; Sebastiani, 1994]. Sebastiani [Sebastiani, 1994] extends $\mathcal{DL}$ by adding two types of probabilities:

- The degree of belief in an assertion γ, denoted $w[\gamma]$ *relop* t, where *relop* could be $=, \leq$, etc., (subjective measure). For example, $w[a : C] = 0.5$ means: the degree of our belief that the individual a is an instance of the concept C, is 0.5. Conditional degree of belief could also be defined. For example, $w[a : C|a : D] = 0.6$ means: the degree of our belief that the individual a is an instance of the concept C knowing that a is an instance of the concept D, is 0.6. The same discussion for subsuming relations, e.g. $w[C \sqsubseteq D] = 0.1$, $w[C \sqsubseteq D|C \sqsubseteq E] = 0.2$.
- Statistical information, denoted $w_{\langle x \rangle}[C]$, where C is a concept, (objective measure). For example, $w_{\langle x \rangle}[C] = 0.3$ means: the probability that a randomly picked individual a is an instance of C, is 0.3. Conditional statistical information could be also defined. For example, $w_{\langle x \rangle}[C|D]$ means: the probability that an individual a is an instance of C knowing that it is an instance of D, is 0.2.

The MIRTL logic [Meghini *et al.*, 1993], besides the above two probabilistic extensions, define a new logic P-MIRTL [Sebastiani, 1994]. In order to give a formal semantic to the new logic P-MIRTL, Sebastiani uses the notion of possible worlds besides the formal semantic used in $\mathcal{DL}$. To define the formal semantic, Sebastiani defines the tuple $M = \{\Delta, I, P_{dom}, P_{int}\}$, where:

- Δ is a set of individuals (domain of interpretation).
- I is a set of interpretations based on Δ.
- P_{dom} is a probability distribution defined on the elements of Δ.
- P_{int} is a probability distribution defined on the elements of I.

The system's degree of belief in a probabilistic statement t in an interpretation $i \in I$, denoted $[t]_{(M,i)}$, is defined as follows:

- Statistical information for concepts: $[w_{\langle x \rangle}C]_{(M,i)} = \sum_{a \in C^i} P_{dom}(a)$ is the sum of all probabilities of those individuals that are instances of C.
- Statistical information for roles: $[w_{\langle x_1, x_2 \rangle}R]_{(M,i)} = \sum_{(a_1,a_2) \in R^i} P_{dom}(a_1) \times P_{dom}(a_2)$ is the sum of all joint-probabilities of those tuples that are instances of R.
- The degree of belief in an assertion: $[w(\gamma)]_{(M,i)} = \sum_{i \in I, (M,i) \models \gamma} P_{int}(i)$ is the sum of probabilities of those interpretations that satisfy the assertion γ.

The non-probabilistic statements of P-MIRTL have the same semantic as in MIRTL. In P-MIRTL, the uncertain IR implication $U(d \rightarrow q)$ is mapped to compute the degree of belief in $d : q$ or $D \sqsubseteq q$ based on: a specific domain, a set of interpretations on that domain, a probability distribution defined on the elements of the domain, and a probability distribution defined on the interpretations.

$$U(d \rightarrow q) = [w(d : q)]_{(M,i)} \quad \text{OR} \quad U(d \rightarrow q) = [w(D \sqsubseteq q)]_{(M,i)}$$

Many other studies to extend $\mathcal{DL}$ by probability exist [Jaeger, 1994, 2006]. However, the probability is not the only way to extend $\mathcal{DL}$ for uncertainty. $\mathcal{DL}$ could be also extended by the notion of possibility [Qi & Pan, 2008]. Possibility-based $\mathcal{DL}$ is more flexible than probabilistic $\mathcal{DL}$, because weights or values in possibility-based $\mathcal{DL}$ could be changed as long as conserving the relative orders with the other weights, which is not the case in the probabilistic $\mathcal{DL}$.

Besides IR, $\mathcal{DL}$ is successfully used in a very close discipline to IR, namely Semantic Web [Berners-Lee *et al.*, 2001]. Since $\mathcal{DL}$ is originally a knowledge representation language, it forms the basis of ontology languages [Baader, 2009], e.g. Web Ontology Language (OWL)[1] and Resource Description Framework Schema (RDFS)[2]. One way or another (mostly manual), documents and queries are transformed to RDF files, where RDF represents the mediator between an ontology, described using OWL or RDFS, and the content of documents and queries. Then, an artificial language (e.g. SPARQL[3]) is used to establish the matching between the RDF files of a document and a query. In fact, on the one hand, automatically transforming documents and queries to RDF files is not an easy task and still an open research area. On the other hand, reasoning or document-query matching is originally binary process (either there is a matching or not), however, there are some attempts to introduce uncertainty [Zhao *et al.*, 2012].

There are several appealing reasons to use $\mathcal{DL}$ in IR. On the one hand, $\mathcal{DL}$ has more expressive power than $\mathcal{PL}$. In other words, $\mathcal{DL}$ enables IR models to represent documents and queries as objects (concepts or individuals) that may have, besides a set of indexing terms, some other properties, e.g. list of authors, publishing date, etc. On the other hand, $\mathcal{DL}$ is originally a knowledge representation language, which means, there are many knowledge bases represented using $\mathcal{DL}$. Therefore, using $\mathcal{DL}$ to represent documents and queries enables us to easily integrate any KB described by $\mathcal{DL}$ into the IR model. However, building IR models based on $\mathcal{DL}$ has some disadvantages. It is not easy to automatically transform documents and queries from their original textual or multi-media form to concepts or individuals in $\mathcal{DL}$. In addition, most expressive families of $\mathcal{DL}$ have unpractical reasoning algorithms[4]. Furthermore, the inference in $\mathcal{DL}$ is originally a binary decision, therefore, $\mathcal{DL}$ in its original form is not suitable for IR. At the same time, extending $\mathcal{DL}$ by some notions of probability or possibility make the extended logic hard to understand and very complex to reason. For example, the two probability distributions P_{dom} and P_{int} in [Sebastiani, 1994] are hard to define.

3.4.5 Models based on Conceptual Graphs

Conceptual Graph (CG) is a bipartite graph of two types of nodes: concepts and conceptual relations. CG is originally proposed as a knowledge representation formalism [Sowa, 1984]. For example, the following structure is a CG:

$$[HUMAN : \#Marie] \rightarrow (MotherOF) \rightarrow [HUMAN : \#John]$$

where '$HUMAN$' is a concept type or the class of all human beings, and '$Marie$' and '$John$' are referents or instances of the class '$HUMAN$'. The structure '$[HUMAN : \#Marie]$' is a

[1] www.w3.org/TR/owl-features/
[2] www.w3.org/TR/rdf-schema/
[3] www.w3.org/TR/rdf-sparql-query/
[4] www.cs.man.ac.uk/ ezolin/dl/

concept, and the node '$MotherOF$' is a conceptual relation type. The previous structure means that, 'Marie', who is a human, is the mother of 'John', who is also a human.

There is a *specification* (sub-type) – *generalization* (super-type) hierarchical relation between concept types, where a concept type C_1 is a sub-type of C_2 *iff* each instance or referent of C_1 is also an instance of C_2, denoted $C_1 \leq C_2$. For example, $FEMALE \leq HUMAN$. There are four main operations that can be defined on CGs:

Copy. A copy of a CG G is another CG G' identical to G.

Restriction. A CG G can be restricted to a CG G' by replacing a concept type (or a relation type) in G with its sub-type. For example, a restriction of the previous CG could be:

$$[FEMALE : \#Marie] \rightarrow (MotherOF) \rightarrow [HUMAN : \#John]$$

Simplification. If a CG G contains a concept related to two identical relations, then one of these relations can be omitted to generate a new CG G' that represents a simplification of G. For example the CG $[C_1 : \#a] \rightarrow (R) \rightarrow [C_2 : \#b] \leftarrow (R) \leftarrow [C_1 : \#a]$ can be simplified to $[C_1 : \#a] \rightarrow (R) \rightarrow [C_2 : \#b]$.

Join. If two CGs G_1 and G_2 have one concept in common, then they can be joined using this concept to build another CG G_3. For example, the CGs $[C_1 : \#a] \rightarrow (R) \rightarrow [C_2 : \#b]$ and $[C_2 : \#b] \rightarrow (R') \rightarrow [C_3 : \#c]$ can be joined in one CG $[C_1 : \#a] \rightarrow (R) \rightarrow [C_2 : \#b] \rightarrow (R') \rightarrow [C_3 : \#c]$.

There is a partial order relation between CGs, where for any two CGs G and G', $G' \leq G$ *iff* G' is derived from G by applying one or more of the previous operations.

CG formalism is equivalent to First Order Logic ($\mathcal{FL}$) [Amati & Ounis, 2000; Chevallet & Chiaramella, 1995, 1998], where each CG G corresponds to a logical sentence $\Phi(G)$, and the partial order relation between CGs corresponds to the validity of material implication [Chevallet & Chiaramella, 1995], as follows: for any two CGs G_1 and G_2[1],

$$G_1 \leq G_2 \Rightarrow \models [\Phi(G_1) \supset \Phi(G_2)]$$

Another operation is also defined on CGs, namely *Projection*. A projection of a CG G on a CG G' is a sub-graph of G', denoted $\pi(G)$, where $\pi(G) \leq G$. In other words, the projection $\pi(G)$ of G on G' is a specialization of G.

Chevallet et al. [Chevallet & Chiaramella, 1995] build an IR model based on CGs. According to them, a document d is a CG, or equivalently, a logical sentence in $\mathcal{FL}$, and a query q is also a CG. The retrieval decision is: d is relevant to q *iff* there exists a projection $\pi(q)$ of q on d, or in other words, *iff* d contains a sub-graph $\pi(q)$ that is a possible specialization of q. From the definition of the projection operation, the retrieval decision is equivalent to check if $d \leq q$ or $\models [\Phi(d) \supset \Phi(q)]$.

The uncertainty of the retrieval decision $U(d \rightarrow q)$ is estimated using Kripke's semantic [Kripke, 1963] or possible world semantic, where a document is a possible world and the accessibility relation between worlds is one of the four main operations on CGs (copy, restriction, simplification, join). The cost of an accessibility relation between two worlds w and w', denoted $P^w(w')$, is related to the operation that causes this transformation from w to w'. Costs are

[1] For more information about the function Φ, see [Amati & Ounis, 2000; Chevallet & Chiaramella, 1995, 1998].

arbitrary assigned to each operation. The uncertainty $U(d \rightarrow q)$ is estimated as in [Nie, 1988], where $U(d \rightarrow q)$ is the cost of the path from d to d_n in which $d_n \leq q$.

CGs is a powerful formalism to express the content of documents and queries. However, it is very difficult to transform the content of documents and queries into CGs, and the projection operation is NP-complete [Chein & Mugnier, 1992].

3.4.6 Models based on Situation Theory

Situation Theory (ST) is a formal framework to model or represent information [Barwise, 1989; Barwise & Perry, 1983]. Instead of studying if a piece of information is true or false, ST studies what makes this piece of information true. According to ST, information tells us that relations hold or not between objects. Therefore, the atomic information carriers are what called *infons*, and an infon is a structure $\langle\langle R, a_1, \ldots, a_n; i\rangle\rangle$ which represents the information that the relation R holds between the objects $a_1, \ldots, a_n$ if $i = 1$, or it does not hold if $i = 0$. A *situation* is a partial description of the world, or it can be defined as a set of infons [Huibers & Bruza, 1994]. A situation s *supports* an infon f, denoted $s \models f$, if f is made true by s, or equivalently, if f can be deduced from the set of infons of s.

ST generalizes a set of situations having common characteristics into a *type of situation*. The notation $s : A$ refers to that the situation s is of the type A. A *constraint* is a relation between two types of situation A and A', denoted $A \Rightarrow A'$. A constraint like $A \Rightarrow A'$ means that the occurrence of a situation $s : A$ implies the existence of a situation $s' : A'$. Uncertainty is represented as a *conditional constraint*, denoted $A \Rightarrow A'|B$, which means that the constraint $A \Rightarrow A'$ is fulfilled under some conditions B, where B itself can be a type of situation.

Lalmas et al. [Lalmas & Rijsbergen, 1993] build an IR model based on ST. According to them, a document d is a situation, and a query q is an infon or a set of infons. The document d is relevant to a query q *iff* d supports q, denoted $d \models q$. The uncertainty of the previous retrieval decision is estimated based on the conditional constraints as follows:

$$U(d \models q) = \begin{cases} 1 & \text{if} \quad d \models q \\ \max_{\{D'|(D \Rightarrow D'|B), \exists d' : D', d' \models q\}} \delta(D, D') & \text{otherwise} \end{cases}$$

where D is the type of d, D' is another type related to D under some conditions B, and $\delta(D, D')$ is defined as follows:

$$\delta(D, D') = \begin{cases} 1 & \text{if} \quad D \Rightarrow D' \\ 0 < \alpha < 1 & \text{if} \quad D \Rightarrow D'|B \\ 0 & \text{otherwise} \end{cases}$$

The function δ is based on the conditions under which it is possible to find another document d' of the type D' where d implies d'. This definition of uncertainty is very close to the definition of Nie [Nie, 1988].

Huibers et al. [Huibers & Bruza, 1994] present two examples, Boolean model and coordination level matching model, on how to build the situations that represent documents and queries, and what the retrieval decision $d \models q$ concretely means. Huibers et al. also use ST to define the *aboutness* relation between a document and a query, in order to build a meta IR model being capable of formally comparing IR models.

The main disadvantage of ST based IR models is the difficulty of automatically building meaningful infons for representing the content of documents and queries. Moreover, uncertainty needs to be defined in a less abstract way in order to build an implementable version of these models.

3.4.7 Models based on Probabilistic Datalog

Datalog is a predicate logic developed in database field. Probabilistic Datalog is an extension of deterministic Datalog using probability [Fuhr, 1995]. The main difference between deterministic and probabilistic Datalog is that probabilistic Datalog defines *probabilistic ground facts*, besides, *deterministic ground facts* that are classical predicates. Probabilistic ground facts have the form αg, where g is a deterministic ground fact (classical predicate), and $0 < \alpha \leq 1$ is the probability that the predicate g is true. Deterministic ground facts are a special case of probabilistic ground facts, where α is always 1.

Ground facts are supposed to be probabilistically independent and mutually disjoint. Assume that $\alpha_1 g_1$ and $\alpha_2 g_2$ are two probabilistic ground facts, where $g_1 \neq g_2$, then the following two probabilistic ground facts are correct:

$$\begin{array}{ll} \alpha_1 \times \alpha_2 & g_1 \wedge g_2 \\ \alpha_1 + \alpha_2 & g_1 \vee g_2 \end{array}$$

Probabilistic Datalog is used in IR [Fuhr, 1995; Lalmas & Bruza, 1998], where a document d is a set of probabilistic ground facts of the form $\alpha\ term(d,t)$ which means that the document d is indexed by the term t and α is the probability that the predicate $term(d,t)$ is true, or equivalently, α is the probability that the document d is about the term t. Fuhr [Fuhr, 1995] extends the previous definition of d, where the terms that index d can be inferred using the predicate $about$ and the following inference rules:

$$\begin{array}{l} about(D,T) : -term(D,T) \\ about(D,T) : -link(D,D') \wedge about(D',T) \end{array}$$

where D, D', T are variables, and the predicate $link(D,D')$ refers to that D and D' are explicitly or implicitly related, e.g. hyperlink. The index of a particular document d is $about(d,T)$. The query q is a Boolean query.

The retrieval decision is defined as an inference rule. However, there are two main forms of the retrieval inference rule based on the connectives between query terms:

- Conjunction $q = t_1 \wedge t_2$:

$$q(D) : -about(D,t_1) \wedge about(D,t_2)$$

  where D is a variable.

- Disjunction $q = t_1 \vee t_2$:

$$\begin{array}{l} q(D) : -term(D,t_1) \\ q(D) : -term(D,t_2) \end{array}$$

The main advantage of using probabilistic Datalog is the clear connection between IR and database, where probabilistic Datalog is an extension of deterministic Datalog that is used in database. However, there still the problem of initially assigning probabilities to ground facts.

3.4.8 Models based on Default Logic

Default logic is used to represent semantic relations between objects, e.g. synonymy, polysemy, etc. It is used in IR to represent some background knowledge or thesauri knowledge. Default logic defines a special inference mechanism called *default rules*, denoted $\frac{\varphi:\beta}{\psi}$, which means that in a particular context if φ is true and $\neg\beta$ can not be inferred then infer ψ.

Positioning is the process of changing a logical sentence using default rules. For example, assume a logical sentence s is $a \wedge b$ where a and b are propositions. Assume the following default rule $\frac{a:\neg c}{e}$ where c and e are also propositions. The positioned sentence s' of s according the previous default rule is $a \wedge b \wedge e$.

Hunter [Hunter, 1995] uses default logic to build an IR model. A document d is a clause, or equivalently, a conjunction of literals and each literal is either a proposition or its negation. Each indexing term corresponds to a proposition. A query q is a logical sentence. The retrieval decision is the material implication where d is relevant to q *iff* $d \supset q$.

Semantic relations between terms are represented through default rules. For example, assume the following default rule:

$$\frac{car \wedge transport : \neg rail}{automobile}$$

This default rule means that if the initial representation of a document d satisfies the two propositions car and $transport$, and the term $rail$ can not be inferred from d, then d can be expanded by the term $automobile$.

The uncertainty of $d \supset q$ is estimated through finding the positioned version d' of d using the set of default rules, where $d' \supset q$. Default logic based IR models have the ability to build *context-dependent* models and to *qualitatively* define the logical uncertainty $U(d \rightarrow q)$. However, these models suffer from some disadvantages. The automatic learning of default rules is not an easy task, and it is error-prone. In addition, there is no quantitative measure of uncertainty.

3.4.9 Conclusions

We presented in the previous subsections a panoramic view of the current state of the art of logic-based IR models. We present the models according to the logics that are used to represent d, q, and $d \rightarrow q$, and also according to the mathematical theories that are used to compute the uncertainty $U(d \rightarrow q)$. Logics are either classical or non-classical logics. Classical logics range from $\mathcal{PL}$ to $\mathcal{FL}$ passing through $\mathcal{DL}$. Uncertainty is computed using a variety of methods, e.g. probability, fuzzy logic, imaging, etc.

Table 3.1 reviews the definitions of: a term t, a document d, a query q, a retrieval decision $d \rightarrow q$, and the uncertainty $U(d \rightarrow q)$, that are used in each logical IR model.

3.5 Exhaustivity and Specificity

Exhaustivity and Specificity were early introduced by Spärck Jones [Jones, 1972]. She talked about *document exhaustivity* which refers to the number of terms the document contains, and

Table 3.1: Logic-based IR models

Models	t	d	q	$d \rightarrow q$	$U(d \rightarrow q)$
Losada & Barreiro 2001	proposition	sentence	sentence	$d \models q$	Dalal's BR ($\circ_D$)
Picard & Savoy 2000	proposition	fact	sentence	$d \supset q$	conditional probability
Nie 1988, 1989	proposition	possible world	sentence	finding a path $d_0 \ldots d_n$	the cost of the path
Nie 1992	proposition	possible world	sentence	-	imaging $P_d(q)$
Crestani & Rijsbergen 1995	possible world	sentence	sentence	-	imaging $P_d(q)$
Nie & Brisebois 1996	proposition	possible world	sentence	finding a path $d_0 \ldots d_n$	fuzzy distance $d(\Diamond^n q)$
Meghini *et al.* 1993	-	individual	concept	assertion $d : q$	-
Sebastiani 1994	-	individual	concept	assertion $d : q$	probability
Chevallet & Chiaramella 1995	-	CG	CG	projection $d \leq q$	the cost of CGs operations
Lalmas & Rijsbergen 1993	-	situation	infon	d supports q $d \models q$	conditional constraint
Fuhr 1995	-	ground facts	Boolean expression	inference rule	probability
Hunter 1995	proposition	clause	sentence	$d \supset q$	positioning

term specificity which refers to the number of documents the term belongs to. According to Spärck Jones, Exhaustivity and Specificity are statistical notions. In this study, we are more concerned with the definition of Exhaustivity and Specificity that was introduced by Nie [Nie, 1988]. According to Nie, Exhaustivity says how much elements of q are mentioned in d, whereas, Specificity says in what level of detail the elements of q are mentioned in d.

Since the logical implication is not symmetric, then when we talk about the implication $d \rightarrow q$, the inverse implication $q \rightarrow d$ intuitively comes to our mind. More precisely, assume a document d and a query q. There are two main questions about d and q come to our mind when we judge about relevance. The first question: is q wholly mentioned in d? normally this question refers to Exhaustivity and denoted by the implication $d \rightarrow q$. The second question, does d concern only q? normally this question refers to Specificity and denoted by the implication $q \rightarrow d$.

Exhaustivity and Specificity are important notions in IR, and most logic-based models are capable of modeling them. For example, Exhaustivity and Specificity are useful in the following situation: assume q is wholly mentioned in two documents d and d'. If d contains information that does not concern q, whereas, the whole information in d' concerns q, in this case, it is prefer-

able for an IR system to rank d' higher than d. Actually, Exhaustivity and Specificity together guaranty this preferable behavior.

According to Nie, Exhaustivity, denoted $d \rightarrow q$, means that q is deducible from d or there is a deductive path from d to q. In other words, Nie represents d as a possible world, then either q is directly true in d, or there is a series of changes that should be done on d to make q true (see Section 3.4.3.2–P.48). Whereas Specificity, denoted $q \rightarrow d$, means that d is deducible from q or there is a deductive path from q to d. That also means, the relation between d and q is not symmetric, where $d \rightarrow q$ is different from $q \rightarrow d$. Nie claims that the retrieval process is not only $d \rightarrow q$ but also $q \rightarrow d$. In addition, Exhaustivity is recall-oriented whereas Specificity is precision-oriented [Losada & Barreiro, 2001].

Many logic-based IR models try to define the two implications $d \rightarrow q$ (Exhaustivity) and $q \rightarrow d$ (Specificity). Losada et al. [Losada & Barreiro, 2001], define Exhaustivity as revising q by d, denoted $q \circ_D d$, whereas, Specificity as revising d by q, denoted $d \circ_D q$. Crestani et al. [Crestani & Rijsbergen, 1995] define Exhaustivity as imaging on d and the logical uncertainty $U(d \rightarrow q)$ becomes $P_d(q)$, whereas, Specificity as imaging on q and the logical uncertainty $U(q \rightarrow d)$ becomes $P_q(d)$.

Exploiting Exhaustivity and Specificity is not restricted to the logic-based IR models. They are used in some statistical models. For example, Crestani [Crestani, 2000] supposes the existence of a function Sim being capable of estimating the semantic similarity between any two terms. He extends the retrieval function to become:

- Query viewpoint: starting from a query and comparing it to a document.

$$RSV_{max(q \triangleright d)}(d, q) = \sum_{t \in q} Sim(t, t^*)\, w_d(t^*)\, w_q(t)$$

  where, $t^* \in T$ is a document term that gives the maximum similarity with t, $w_d(t^*)$ is the weight of t^* in d, and $w_q(t)$ is the weight of t in q.

- Document viewpoint: starting from a document and comparing it to a query.

$$RSV_{max(d \triangleright q)}(d, q) = \sum_{t \in d} Sim(t, t^*)\, w_d(t)\, w_q(t^*)$$

Crestani claims that $q \triangleright d$ refers to Exhaustivity and $d \triangleright q$ refers to Specificity.

Exhaustivity and Specificity are also exploited in structured documents and passage retrieval, where IR systems try to find a document d that answers a query q (Exhaustivity), and within d, systems try to find the most appropriate or precise component or passage (Specificity) [Chiaramella *et al.*, 1996].

The two notions, Exhaustivity and Specificity, have been studied in the state of the art but mostly from a theoretical point of view.

3.6 Lattice-Based IR Models

There is a potential relation between some logics and lattice theory, where the logical implication becomes a partial order relation. In this study, for example, we use lattice theory[1] to

[1]For more information about lattice theory, see (Appendix B–P.175).

formally define the logical uncertainty $U(d \rightarrow q)$. In literature, there are many studies that use lattice theory to build IR models, however, lattice theory is used in a very different way of how we use it in this study. We think it will be a good idea to briefly present the main lattice-based IR models.

In this section, we talk about using lattices in IR field. We divide our presentation into two main categories: First, using lattices without *Formal Concept Analysis* (FCA), and second, using FCA.

3.6.1 Lattices Based IR

One of the earliest studies that exploit the *Lattice* algebraic structure in IR is Mooers' study [Mooers, 1958]. Mooers defines two lattices:

- The lattice of all possible answers $L = (2^D, \cap, \cup)$, where D is the set of documents and 2^D is the power set of D. Each node x of L represents one possible answer (set of documents) that an IR system could retrieve.
- The lattice of all possible queries P. Mooers builds three different variants of P according to the following situations:
  - the query q is simply a set of terms. Here, $P = (2^T, \cap, \cup)$ where T is the set of indexing terms and 2^T is the power set of T. In this case, P is obtained by taking the product of some *elementary lattices*, where each term $t \in T$ corresponds to one elementary lattice L_t. The lattice L_t contains two nods: the infimum $\perp$ and t, ordered as follows $\perp \leq t$.
  - q is a Boolean expression of terms and the two logical connectors: $\wedge$ and $\neg$. Here also, P results from the product of some elementary lattices, where each term t corresponds to one elementary lattice L_t. The lattice L_t contains four nodes: $\perp$, t, $\neg t$, and $\top$, equipped by the following partial order relations: $\perp \leq t \leq \top$ and $\perp \leq \neg t \leq \top$.
  - the terms are not independent and there is a hierarchical relation between them, e.g. 'shoes' and 'clothes' where 'shoes' is *a kind of* 'clothes'. Here also, P is a product of some elementary lattices, where each elementary lattice corresponds to one possible chain like: $\perp \leq$ 'clothes' $\leq$ 'shoes'.

Even though the retrieval process is not clearly defined in [Mooers, 1958], however, in principle, the retrieval process is a transformation from a node x in P (a query) to a node y in L (a set of documents). Mooers defines two variants of this transformation: 1- each document $d \in y$ must be exactly indexed by the set of terms x, 2- each document $d \in y$ can be indexed by any super-set of terms x' where $x \subseteq x'$.

The main disadvantage of Mooers' model is that it is impractical. However, Priss presents a new IR system using lattices [Priss, 2000]. Counter to Mooers, who uses lattices to represent the content of documents and queries (*data-driven approach*), Priss uses lattices to represent the semantic content of a knowledge base, more precisely, Priss exploits the relationships between the terms of a domain knowledge (*faceted thesaurus-driven approach*).

According to Priss, the IR system consists of a set of documents D, a query language Q, a faceted thesaurus T/F, and a set of concepts C. The faceted thesaurus T/F consists of a set

of terms T partitioned into a set of facets or viewpoints F. Each facet is a lattice, where each node in this lattice corresponds to a term. A term can be a word or a phrase. The lattice of a particular facet reflects some conceptual relations between terms, e.g. 'C++' is a 'multi-purpose programming language' and any 'multi-purpose programming language' is a 'programming language'. The set of concepts C contains *simple* and *complex* concepts, where each term in a particular facet is a simple concept, and a complex concept is a composition of terms from different facets (terms from one facet can not be composed). A document $d \in D$ is indexed by a set of concepts C_d, where C_d contains at most one concept per facet. Any query is a Boolean expression of concepts, where $Q = (C, \wedge, \vee, \neg)$. Priss defines two types of search:

- *Intra-facet* search, where queries consist of simple concepts of the same facet. Here, Priss also distinguishes between two methods of search: 1- *exclusive*: retrieves an exact concept, and 2- *inclusive*: retrieves an exact concept besides the more specific and more general concepts.
- *Inter-facet* search, where queries consist of simple concepts of more than one facet.

In all cases, the retrieval process is the classical Boolean retrieval.

3.6.2 Formal Concept Analysis Based IR

3.6.2.1 Introduction to Formal Concept Analysis

Originally, Formal Concept Analysis (FCA) is a sub-field of applied mathematics [Wille, 1982, 2005; Wolff, 1993]. Wille [Wille, 2005] explains the potential relation between FCA and the philosophical logic of human thought. Philosophically, concepts are the basic units of human thought. Any *concept* can be defined through its *extension*, which contains all *objects* that belong to this concept, or through its *intention*, which includes all *attributes* or properties that apply to all objects of the extension (see Section 2.2–P.18). There are always relationships between concepts, where the most important relationship is the *subconcept-superconcept* relationship. FCA is used to mathematically talk about concepts, attributes, objects, intention, and extension.

A *formal context* is defined as the structure $K = (G, M, I)$ where, G is a set of *formal objects*, M is a set of *formal attributes*, I is a binary relation $I \subseteq G \times M$, and gIm or equivalently $(g, m) \in I$ is read: *the object* g *has the attribute* m. There are two derivation operators:

- for any set of objects $X \subseteq G$, there is a set of attributes $X^I \subseteq M$ defined as follows:

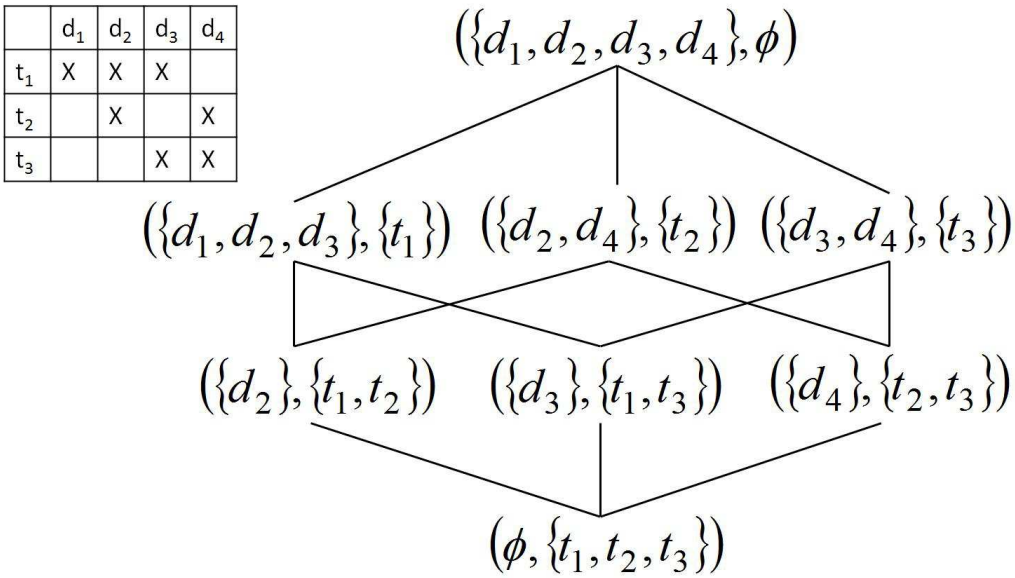
$$X^I = \{m \in M | \forall g \in X, gIm\}$$

  or equivalently X^I is the set of attributes that can be applied to all objects of X.
- for any set of attributes $Y \subseteq M$, there is a set of objects $Y^I \subseteq G$ defined as follows:

$$Y^I = \{g \in G | \forall m \in Y, gIm\}$$

  or equivalently Y^I is the set of objects that have at least all attributes of Y.

Figure 3.1: A term-document adjacency matrix and its corresponding concept lattice.



A *formal concept* of a *formal context* K is defined as a pair (X, Y) where, $X \subseteq G$ is the *extent* of the concept, $Y \subseteq M$ is the *intent* of the concept, $X = Y^I$, and $Y = X^I$. The subconcept-superconcept relationship is mathematically defined as follows:

$$(X_1, Y_1) \leq (X_2, Y_2) \Leftrightarrow X_1 \subseteq X_2 \Leftrightarrow Y_2 \subseteq Y_1$$

The set of all formal concepts of K together with the previous partial order relation forms a *complete lattice*, where:

- The infimum $\perp$ is: $\bigwedge_i (X_i, Y_i) = \left(\bigcap_i X_i, (\bigcup_i Y_i)^{II}\right)$
- The supremum $\top$ is: $\bigvee_i (X_i, Y_i) = \left((\bigcup_i X_i)^{II}, \bigcap_i Y_i\right)$

3.6.2.2 FCA Based Models

FCA is used in IR field in many different ways. The most direct way is to manipulate documents as objects and terms as attributes. In other words, defining the term-document adjacency matrix to be a formal context. In this formalism, a formal concept (X, Y) means that X is a set of documents, Y is a set of terms, and every document in X is at least indexed by all terms of Y. Figure 3.1 shows an example of a possible term-document adjacency matrix and its corresponding concept lattice, where $\times$ sign in the table cell (t_i, d_j) means that the document d_j is indexed by the term t_i.

Messai et al. [Messai *et al.*, 2006] exploit the previous mapping between FCA and the term-document matrix to define a FCA-based IR system. According to Messai et al., a query q is a pair $(\{x\}, \{x\}^I)$ where $\{x\}^I$ is a set of attributes, or equivalently, a set of terms, and $\{x\}$ is a *dummy object*. They first extend the term-document matrix by the pair $(\{x\}, \{x\}^I)$, and then they rebuild the concept lattice. To retrieve the relevant documents of the query $(\{x\}, \{x\}^I)$,

they scan the concept lattice starting from the *pivot concept* $P = (\{x\}^{II}, \{x\}^{I})$. The relevant documents of the query $q = (\{x\}, \{x\}^{I})$ are the objects (documents) in the extent of P, i.e. $\{x\}^{II}$, and the objects in the extents of all superconcepts of P. Codocedo et al. [Codocedo *et al.*, 2012] also include the objects in the extents of *close concepts*, where any two non-comparable concepts (X_1, Y_1) and (X_2, Y_2) are close *iff* there exists another concept (X_3, Y_3) such that $(X_3, Y_3) \leq (X_1, Y_1)$ and $(X_3, Y_3) \leq (X_2, Y_2)$. This idea of closeness is proposed to overcome some well-known IR problems like term-mismatch.

However, defining the term-document adjacency matrix as a formal context has two main disadvantages:

- the corresponding concept lattice will be very large. Cheung et al. [Cheung & Vogel, 2005] present a way to reduce the concept lattice that corresponds to a term-document matrix. They depend on the Singular Value Decomposition theory.
- the previous term-document matrix is a binary matrix, and FCA is originally built depending on binary formal contexts, i.e. either an object has an attribute or not. However, this type of binary relations is not the appropriate ones to represent the relationship between terms and documents. Djouadi [Djouadi, 2012] thus proposes to use fuzzy formal concepts to build the concept lattice, and then to build a more realistic IR system.

As we mentioned, the concept lattice, which corresponds to a term-document matrix, is a very large lattice. In addition, transforming the term-document matrix into a concept lattice results in an IR system very similar to a simple keyword-based IR system. To overcome these problems, Rajapakse et al. [Rajapakse & Denham, 2006] propose another approach. Each document or query is represented by a concept lattice. Formal concepts are extracted from the text using a set of ad-hoc rules, where both objects and attributes are terms. The basic unit of document representation is an object-attribute pair, or equivalently, term-term pair called a *unit-concept*. The direct matching between the document concept lattice and the query concept lattice depends on the number of shared unit-concepts between them. Rajapakse et al. also introduce the notion of partial matching. Besides that, they depend on the simple keyword matching in the cases where there is no shared unit-concepts between a document and a query.

For more exhaustive picture about the potential capabilities of FCA, that could be exploited in IR, we refer the reader to the paper of Carpineto et al. [Carpineto & Romano, 2005].

3.7 Conclusion

In this chapter, we tried to present a broad picture of the current state of the art of logic-based IR models. These models have the following general definition: assume a logic $\mathcal{L}$, assume that a document d is a logical sentence in $\mathcal{L}$, and a query q is also a logical sentence in $\mathcal{L}$. The document d is relevant to q *iff* d logically implies q, denoted $d \rightarrow q$. The implication operation '$\rightarrow$' is not forcibly a part of the formal language of the logic $\mathcal{L}$. The implication $d \rightarrow q$ is generally a binary decision. However, IR is intrinsically uncertain process. Therefore, an uncertainty measure should be defined to estimate the degree to which d implies q, denoted $U(d \rightarrow q)$.

The previous general definition of logical IR models is concretized using different families of logics, distributed on classical and non-classical logics, e.g. modal logic. The classical ones

range from Propositional Logic $\mathcal{PL}$ to First-Order Logic $\mathcal{FL}$ passing through Description Logic $\mathcal{DL}$. Table 3.1, depicts, for each concrete logic-based IR model, the definitions of the four main components of any IR model: document, query, retrieval decision, and uncertainty.

Actually, each model has its advantages and disadvantages. However, in general, logic-based IR models have some common disadvantages. First, most logic-based IR models are far from being operational systems. Most models stay rather theoretical models than practical ones. That maybe happen because the reasoning algorithms related to these models are computationally complex. Actually, there is a tradeoff between the expressive power of logics and the computational complexity of related reasoning algorithms. The more expressive logics are, the more complex their algorithms are. Many studies [Koller *et al.*, 1997; Sebastiani & Straccia, 1991], try to find the optimal balance between the expressive power and the complexity of algorithms. Second, the notion of uncertainty is usually an external component different from the logic itself. Most logics have strict value reasoning (True or False), and this is not suitable for IR. Thus, a new module should be added to the logic in order to simulate IR uncertainty. Normally, uncertainty is artificially added to the logic, and that makes the logic very complex framework for IR and the computational complexity also becomes more complex. Third, some models do not give a precise definition of the implication '$\rightarrow$', instead of that, they directly deal with uncertainty $U(d \rightarrow q)$. For example, modal logic based models [Crestani & Rijsbergen, 1995; Nie, 1988, 1992; van Rijsbergen, 1986]. Finally, in some models, document and query representations are very hard to obtain. For example, conceptual graph based models [Chevallet & Chiaramella, 1995], and situation theory based models [Lalmas & Rijsbergen, 1993].

Logic-based IR models have also some common advantages. Logical IR models provide a good framework for understanding and describing the main IR components (document, query, matching), and for the theoretical comparison of IR models, e.g. meta-models [Huibers & Bruza, 1994]. In addition, defining documents and queries as logical sentences and the retrieval process as an inference, makes logic-based IR models very general and flexible. Actually, some logical models are capable of reproducing some classical IR models. Moreover, logical models are very flexible because document and query definitions and the inference mechanism can be concretized in several ways. Furthermore, most logical IR models are capable of explicitly or implicitly integrating external knowledge into an IR model, e.g. default logic [Hunter, 1995].

As we mentioned in (Chapter 1–P.3), formal logics are useful tools to formally represent and integrate knowledge in IR models, and also useful tools to reproduce the inferential nature of the retrieval process. These are the main motivations beyond our choice to study logic-based IR models and to build a new model.

In this chapter, we also talked about the two IR notions: Exhaustivity and Specificity, which imply that the relation between d and q is not symmetric, where comparing d to q, in principle, is different from comparing q to d. The chapter also covers some lattice-based IR models, where they are mainly based on FCA.

Part III

CONTRIBUTION

Chapter 4

Revisiting the IR Logical Implication

4.1 Introduction

Logic-based Information Retrieval (IR) models represent the retrieval decision by a logical implication $d \rightarrow q$, where d represents a document and it is a set of logical sentences in a particular logic, and q represents a query and it is also a logical sentence in the same logic of d. In the logical IR field, there is a well-accepted assumption and claim that the implication '$\rightarrow$' is different from the classic material implication '$\supset$', and the material implication is not suitable for IR [Nie, 1988; van Rijsbergen, 1986]. However, each logic-based IR model presents its own definition of $d \rightarrow q$, which mainly depends on the used logic.

We here postulate that the retrieval decision between a document d and a query q can be represented through the validity of material implication, denoted $\models d \supset q$. Hereafter, we discuss the validity of our hypothesis through three main points:

- Showing that the argumentation beyond the need for a non-classical-implication assumption was partially inconvenient (Section 4.3).
- Discussing the *validity* of material implication vs. its *truth* (Section 4.4).
- Explaining the meaning of *false* documents (Section 4.5).

4.2 Propositions vs. Indexing Terms

We use Propositional Logic ($\mathcal{PL}$) as the underlying logic[1]. Therefore, any logical sentence s is a logical sentence in $\mathcal{PL}$, and it is built based on a set of atomic propositions Ω. The set of atomic propositions is the set of indexing terms. In other words, every term a is an atomic proposition.

A term a *is true* in a specific document d means that a indexes d, or equivalently, d is about a [Sebastiani, 1998]. The phrase 'a indexes d' does not simply mean that the term a appears in d, although in most cases this simplification is taken into account when implementing the model, it means that one of the topics covered in d is a. For example, assume a document d talking about trees and d contains many images of trees, and suppose that d contains many phrases of

[1] See (Appendix C–P.185) for more information about the formal language of $\mathcal{PL}$ and its formal semantic.

the form '*see image x*', although d contains the term '*image*' it does not talk about the concept of images, and then d should not be indexed by the term '*image*'. Conversely, assume d talks about several species of trees without using the term '*tree*'. Although the term '*tree*' does not explicitly appear in d but it is preferable to index d using the term '*tree*'.

A term a *is false* in d means that d contains an explicit or implicit information or knowledge saying that d must not be indexed by a, or d is not about a.

For the other terms, where there is a doubt if d is about them or not, there are two choices: either considering them false in d (close world assumption), or they can be true or false (open world assumption)[1]. Documents and queries are logical sentences built based on terms.

Example 4.1. *Assume the document* d *is* 'Karam lives in an apartment in Grenoble'. *The document is about a person* 'Karam', *and then the corresponding proposition* a_1 *of the term* 'Karam' *must be true in* d. *In the same manner, if we suppose that* a_2, a_3, a_4, a_5 *are the corresponding propositions of the terms* 'live', 'apartment', 'Hong Kong', 'Grenoble', *respectively, then* d *possibly becomes the following logical sentence:* $d = a_1 \wedge a_2 \wedge a_3 \wedge \neg a_4 \wedge a_5$, *where* $\neg a_4$ *refers to that the person* 'Karam' *has nothing to do with* 'Hong Kong'. *Now assume that* a_6 *is the corresponding proposition of the term* 'France'. *By knowing that* 'Grenoble' *is a french city, then we do not exactly know if* a_6 *must be true in* d *or false, and hence* d *could be* $d = a_1 \wedge a_2 \wedge a_3 \wedge \neg a_4 \wedge a_5 \wedge a_6$ *or* $d = a_1 \wedge a_2 \wedge a_3 \wedge \neg a_4 \wedge a_5 \wedge \neg a_6$.

Furthermore, by knowing that 'flat' *and* 'apartment' *are two synonymous terms then* d *can be rewritten as* $d = a_1 \wedge (a_2 \vee a_2') \wedge a_3 \wedge \neg a_4 \wedge a_5$, *where* a_2' *is the proposition that corresponds to the term* 'flat'. *Of course, there are many other ways to express the content of* d.

Transforming a document d to a logical sentence is still problematic. Some studies [Mooers, 1958] argue that it is meaningless to use disjunction ($\vee$) between terms, and the only two connectives needed are conjunction ($\wedge$) and negation ($\neg$). Therefore, since the automatic recognition of negation within text is a very difficult practice, most IR models assume that the logical sentence that represents d is the conjunction of the terms that appear in d and the conjunction of the negative form of the terms that do not appear in d. In other words, all terms that appear in d must be true and the other terms must be false (close world assumption).

4.3 Do Really We Need a Non-Classical Implication?

All definitions of the implication $d \rightarrow q$, depicted in [Chevallet & Chiaramella, 1998; Chiaramella & Chevallet, 1992; Crestani & Lalmas, 2001; Losada & Barreiro, 2001; Meghini *et al.*, 1993; Nie, 1988; van Rijsbergen, 1986], agree that $d \rightarrow q$ can only be evaluated (is true or false) in the interpretations where the antecedent d is true and in those interpretations if the consequent q is also true then $d \rightarrow q$ is true. In other words, the evaluation space of $d \rightarrow q$ is restricted to the models of d, $M(d)$. Conversely, the evaluation space of the material implication $d \supset q$ contains all possible interpretations, or in other words, $d \supset q$ is evaluable in any interpretation.

The need for a non-classical implication $d \rightarrow q$ is first introduced by van Rijsbergen [van Rijsbergen, 1986]. In fact, the discussion of van Rijsbergen is built upon two main points:

[1] See (Appendix C–P.185) for more information about the open and close world assumptions.

- He uses probability to estimate the uncertainty of logical implications. He assumes that it is possible to estimate the uncertainty of the non-classical implication $U(d \rightarrow q)$ through a conditional probability:

$$U(d \rightarrow q) = P(d \rightarrow q) = P(q|d)$$

  He shows that $U(d \rightarrow q)$ gives different results comparing to $U(d \supset q)$, where:

$$U(d \supset q) = P(d \supset q) = P(\neg d \vee q)$$

  He intuitively assumes that the conditional probability is the right one for IR. He says:

  > "*I would maintain that the conditional probability interpretation in the context of Information Retrieval is the right one.*"

- He then tries to validate a specific criterion, which must be valid when using probability to estimate uncertainty. He exactly says:

  > "*There is another reason why a conditional must not be identified with the material implication in logic. When using probabilistic inference, we want to ensure that the following soundness criterion holds. It is impossible for the premises of an inference to be probable while its conclusion is improbable. [...]. In our example, whenever* $\neg A$ *is true,* A *will be false and hence* $A \supset B$ *will be true, independent of* B*'s truth value. If we identified* $A \rightarrow B$ *with* $A \supset B$*, then such an inference could easily violate the soundness criterion.*"

There are two restrictions in the discussion of van Rijsbergen. First, he restricts uncertainty to probability, even there are other more relaxed theories for estimating the uncertainty of a logical implication. In general, there is no rule saying that the uncertainty of a logical implication must be estimated through probability. Second, even using probability, van Rijsbergen does not formally justify replacing $U(d \rightarrow q)$ by $P(q|d)$ and replacing $U(d \supset q)$ by $P(\neg d \vee q)$, he only gives an example. Let us reconsider the assumption of van Rijsbergen, which states that the conditional probability $P(q|d)$ is the correct choice for IR. We will show, within (Chapter 5–P.77) and as a direct result of our model, that the uncertainty $U(\models d \supset q)$ can be replaced by $P(q|d)$ [Abdulahhad *et al.*, 2013a].

The other motivation under the assumption that the material implication is not suitable for IR is the *false document* problem. This problem comes from the formal definition of material implication, where if d is false then $d \supset q$ is true. By representing the retrieval decision through the truth of $d \supset q$, where d is relevant to q *iff* $d \supset q$ is true in an interpretation, the false document problem appears, because whatever q is, a false document is a possible answer. Actually, we think this is a *superficial* problem, because, on the one hand, discussing the meaning of '*d is false*' shows that false documents represent some type of documents that we do not normally face in our daily systems. On the other hand, a false document makes the implication $d \supset q$ *true* but not *valid*, which means, when d is false in an interpretation δ then $d \supset q$ is true under δ, denoted $\{\delta\} \models d \supset q$, but that does not mean $d \supset q$ is valid (always true under any interpretation). Therefore, the false document problem can be avoided by representing the retrieval decision through the validity of material implication $\models d \supset q$ instead of only its truth in an interpretation [Sebastiani, 1998].

4.4 The Validity of Material Implication

A valid logical sentence s is a true sentence in all interpretations, denoted $\models s$. Let us now study the validity of the IR material implication $\models d \supset q$. The implication $d \supset q$ is valid in only three cases:

- Unsatisfiable documents $\not\models d$ (d is always false), where if d is unsatisfiable then $d \supset q$ is valid.
- Valid queries $\models q$ (q is always true), where if q is valid then $d \supset q$ is also valid.
- The general case $M(d) \subseteq M(q)$, where $d \supset q$ is valid *iff* in any interpretation δ if d is true $\{\delta\} \models d$ then q is also true $\{\delta\} \models q$ (Theorem C.1–P.188).

First, unsatisfiable documents $\not\models d$ represent either empty documents or documents containing contradictions, e.g. $a \wedge \neg a$ where a is an indexing term. In both cases, unsatisfiable documents are a type of documents that we do not normally face in real IR systems, or they are special cases that can be separately manipulated. Second, valid queries represent always-true user needs. A query like '*retrieve documents that talk about trees or other things different from trees*', or mathematically $a \vee \neg a$, is an example of valid queries. It is correct, from the system point of view, to retrieve all documents as an answer to a valid query.

Finally, the general case corresponds to the definition of non-classical implication $d \rightarrow q$. Van Rijsbergen [van Rijsbergen, 1986] exactly says about the non-classical implication:

> "*Let s be a partial description of a document —this might be a set of sentences, or just a single index term— q being a request. In deciding whether to retrieve a document we would need to evaluate $s \rightarrow q$, that is, whether $s \rightarrow q$ is true or not. If s is true in a document d then $s \rightarrow q$ is true providing q is true. If s is not true in a document then we go to the nearest document d' to d in which it is true and consider whether q is true. If q is true in d' then $s \rightarrow q$ is true in d, otherwise it is false.*"

Chiaramella et al [Chiaramella & Chevallet, 1992], in their discussion of $d \supset q$, assume that d is a *fact*, or in other words, they evaluate $d \supset q$ only in the cases where d is true.

Through the previous discussion, we illustrate, in a descriptive manner, that the non-classical implication $d \rightarrow q$ is a special case of the material implication $d \supset q$. Here, we also present a more formal illustration showing that $d \rightarrow q$ is a special case of $d \supset q$, as follows:

Point 1. $M(d \rightarrow q)$ vs. $M(d \supset q)$

Based on the truth table, the material implication $d \supset q$ is true, when either: both d and q are true, or d is false. Therefore, the set of models of $d \supset q$ is $M(d \supset q)$:

$$M(d \supset q) = [M(d) \cap M(q)] \cup M(\neg d)$$

According to the definition of the non-classical implication $d \rightarrow q$, which is presented in [Chiaramella & Chevallet, 1992; van Rijsbergen, 1986], $d \rightarrow q$ is true in a particular interpretation *iff* d and q are true in that interpretation, or equivalently, if d is true in a particular interpretation δ then $d \rightarrow q$ is true in δ *iff* q is also true in δ. Thus, the set of models of $d \rightarrow q$ is $M(d \rightarrow q)$:

$$M(d \rightarrow q) = M(d) \cap M(q)$$

We can see that $d \rightarrow q$ is a special case of $d \supset q$, where $M(d \rightarrow q) \subseteq M(d \supset q)$.

Point 2. $(\models d \rightarrow q)$ vs. $(\models d \supset q)$
Based on (Theorem C.1–P.188), we know that:

$$[\models d \supset q] \Leftrightarrow [M(d) \subseteq M(q)]$$

The implication $d \rightarrow q$ is only evaluable in the cases where d is true or in $M(d)$. Furthermore, the implication $d \rightarrow q$ is also equivalent to the set inclusion between models:

$$[M(d) \models d \rightarrow q] \Leftrightarrow [M(d) \subseteq M(q)]$$

because

- $M(d) \models d \rightarrow q$, means that every model of d is also a model of $d \rightarrow q$, which in turn according to [Chiaramella & Chevallet, 1992; van Rijsbergen, 1986] means that q must be true, and consequently $M(d) \subseteq M(q)$.
- $M(d) \subseteq M(q)$, means that when d is true then q is also true, and consequently $d \rightarrow q$ is true.

On the one hand, from *Point 1* and *Point 2*, we can see that: $d \rightarrow q$ is a special case of $d \supset q$. On the other hand, we show that the cases when d is false are either trivial or do not affect the behavior of the validity of material implication. Therefore, we claim that $\models d \supset q$ is a suitable choice to represent the retrieval decision.

4.5 What does 'd is false' Mean?

To discuss 'd *is false*', we have three distinctive cases:

- Unsatisfiable documents $\not\models d$, which means that d is false in all possible interpretations, or in other words, either d contains a contradiction of the form $(\ldots a \wedge \neg a \ldots)$ where a is an indexing term, or d is an empty document. In both cases, d is trivial and we do not normally face this case in real IR systems.
- Documents about nothing or uninformative documents $d = \neg a_1 \wedge \cdots \wedge \neg a_n$, mathematically $\{\emptyset\} \models d$, or in other words, the only model of d is the empty set where all terms must be false. In this case, it is possible to find an interpretation $\delta \neq \emptyset$ where d is false $\{\delta\} \not\models d$, and hence the implication $d \supset q$ is true in this interpretation $\{\delta\} \models d \supset q$, but that does not mean $d \supset q$ is valid, or equivalently $\models d \supset q$. In all interpretations, except the interpretation $\delta = \emptyset$ where all propositions are set to false, the uninformative document d is false, and then $d \supset q$ is true, whereas in $\delta = \emptyset$, d is true, therefore, the implication $d \supset q$ to be valid, the query q must be also true in $\delta = \emptyset$. The only query that satisfies this condition is the uninformative query. Therefore, for an uninformative document d, the implication $d \supset q$ is valid only if the query q is also uninformative. In this case it is reasonable to retrieve an uninformative document as an answer to an uninformative query [Sebastiani, 1998].

- The general case is when d is about something, or mathematically, there is at least an interpretation $\delta \neq \emptyset$ where $\{\delta\} \models d$. In principle, the goal of logic-based IR systems is to evaluate the logical implication $d \rightarrow q$, more precisely, they start from d as a starting point and try to check if it implies q or not. To evaluate $d \rightarrow q$, there is an evaluation space which is the set of all possible interpretations Δ. Normally, Δ is a very large set and IR systems do not check the truth of $d \rightarrow q$ in all interpretations Δ, instead of that, they reduce the evaluation space Δ to a subset of interpretations $M(d) \subseteq \Delta$, where $M(d) \models d$ is the set of models of d, and in most of the cases $M(d)$ only contains one interpretation. They do that because they start the search from d, so there is no need to check the interpretations where d is false. Furthermore, for a particular interpretation $\delta \in \Delta$, $\{\delta\} \not\models d$ means $\{\delta\} \models d \supset q$ but not $\models d \supset q$, or it means that $d \supset q$ is true in δ but not in all interpretations Δ. In other words, $\{\delta\} \not\models d$ means that the material implication $d \supset q$ is true in δ, but it does not mean that $d \supset q$ is valid, which is our goal. The validity $\models d \supset q$ means either that d is unsatisfiable $\not\models d$, or that when d is true then q must be also true.

From the previous discussion, we illustrate that $\not\models d$ is an unrealistic case, and also the validity of material implication does not suffer from the false document problem. Therefore, we think that modeling the retrieval decision through the validity of material implication is an appropriate choice for IR. Our previous discussion is compatible with the conclusions of Sebastiani [Sebastiani, 1998].

4.6 Conclusion

We show that the argumentation about the need to a non-classical implication is not rigid, because van Rijsbergen restricts the uncertainty of implication U to conditional probability, and also he considers the truth of material implication instead of its validity to finally decide the unsuitability of material implication for IR. We also show that the false document problem is a superficial problem, where false documents either represent trivial documents or do not affect the validity of material implication. Moreover, we illustrate that the non-classical implication $d \rightarrow q$ is a special case of the material implication $d \supset q$. After the previous discussion our main hypothesis is:

Hypothesis 4.1. *The material implication $d \supset q$ is an appropriate implication for modeling the retrieval decision in logic-based IR models. Precisely, the retrieval decision is the validity of material implication $\models d \supset q$, where d represents a document and it is a logical sentence in a specific logic, and q represents a query and it is also a logical sentence in the same logic.* □

Informally, d is an answer to q *iff* the material implication $d \supset q$ is valid (not only true in a particular interpretation but always true in any interpretation). Choosing *validity* instead of *truth* enables us to reason about relevance in a form-based way comparing to the content-based way of truth [Sebastiani, 1998].

It is also possible to introduce, in an intuitive manner, a way of estimating the uncertainty of an implication. By knowing that $[\models d \supset q] \Leftrightarrow [M(d) \subseteq M(q)]$, we can simply claim that:

$$U(\models d \supset q) = \frac{|M(d) \cap M(q)|}{|M(d)|}$$

The intuitive meaning of this formula could be that the degree to which d and q are compatible, or how many system (people) could assign the same interpretation to both d and q. We will, later in this thesis, introduce a more formal measure for estimating the uncertainty.

Chapter 5

A New Logic and Lattice Based IR Model

5.1 Introduction

In (Chapter 1–P.3), we presented the motivations of using formal logics in Information Retrieval (IR). Actually, formal logics are very important tools to represent knowledge, and then to formally integrate it in IR models. In addition, formal logics are also capable of simulating and reproducing the inferential nature of the retrieval process. In (Chapter 3–P.39), we presented a panoramic view of the current state of the art of logical IR models, and we illustrated their main shortcomings. In this chapter, we present our contribution, which is mainly to propose a logic and lattice based IR model being capable of overcoming some disadvantages of current logical models.

Generally, IR is the process of retrieving among a set of documents, the documents that are *likely* relevant to a query. Logic-based IR models represent documents and queries as logical sentences, the retrieval decision as an implication $d \rightarrow q$, and the ranking mechanism by the degree to which a document logically implies a query $U(d \rightarrow q)$.

In this chapter, we propose a logic-based IR model. The underlying mathematical frameworks are Propositional Logic ($\mathcal{PL}$) and lattice theory. A document d (or a query q) is a logical sentence written in Disjunctive Normal Form (DNF), and the retrieval decision is the validity of material implication, in other words, d is relevant to q *iff* the material implication $d \supset q$ is valid, denoted $\models d \supset q$ (Hypothesis 4.1–P.74).

We know that there is a potential relation between $\mathcal{PL}$ and lattices (Appendix C–P.185). Accordingly, the IR process within lattice framework becomes: d (or q) is one or several nodes in a predefined lattice, and the retrieval decision is equivalent to the partial order relation that is defined on this lattice. The uncertainty of $\models d \supset q$, which is now equivalent to the partial order relation of the lattice, is estimated through the degree of inclusion function Z, which quantifies the partial order relation and computes the degree to which a node in a lattice contains another node. The Z function has been introduced by Knuth [Knuth, 2005].

It is suggested, especially in case of unfamiliarity with lattices and $\mathcal{PL}$, to read (Appendices B–P.175 and C–P.185) for more information about the following mathematical notions: lattices as algebraic structures, quantifying the partial order relations of lattices, the formal language and semantic of $\mathcal{PL}$, and the very initial relation between lattices and the formal semantic of $\mathcal{PL}$.

This chapter is organized as follows: In section 2, we redefine the relation between $\mathcal{PL}$ and lattices through proposing an intermediate representation for logical sentences. We also show that, using the intermediate representation, it is possible to transform checking the validity of material implication to a series of simple set-inclusion checking. In section 3, we propose an IR model based on the intermediate representation of $\mathcal{PL}$ logical sentences. The model defines in detail the main four components of any IR model: a document, a query, the retrieval decision, and the ranking mechanism. Section 4 is dedicated to discuss some valuable conclusions of our proposed model, including: the new formalism of van Rijsbergen's assumption about estimating the uncertainty $U(d \rightarrow q)$ through the conditional probability $P(q|d)$, the new approach to deal with the two abstract notions Exhaustivity and Specificity, and finally showing that our model is a general framework being capable of reproducing most classical IR models. We conclude in section 5.

5.2 From Propositional Logic to Lattices: A Mathematical Perspective

The formal interpretation or semantic of $\mathcal{PL}$ maps each logical sentence s to a set of models $M(s)$. A Boolean algebra $\mathcal{B}_M$ is then built upon this formal semantic (Theorem C.2–P.188), where each logical sentence s corresponds to a particular node $M(s)$ in $\mathcal{B}_M$, and the partial order relation defined on $\mathcal{B}_M$ is equivalent to the validity of material implication $\supset$ (Theorems C.1 & C.2–P.188).

On the one hand, the nodes of $\mathcal{B}_M$ are not simple, where each node of $\mathcal{B}_M$ is a set of sets of elements. Although it is possible to build a simpler lattice [Knuth, 2003, 2005], but in that case there will not be a direct mapping between the partial order relation and the material implication, which is very important for IR. Moreover, IR needs simplicity in order to build efficient models, where IR notions like documents and queries are generally modelled in a very simple way e.g. bag of terms.

On the other hand, given a logical sentence s, its set of models $M(s)$ is computable. However, from a set of interpretations it is almost impossible to know the corresponding sentence, because each set of models $M(s)$ models a set of sentences logically equivalent to s. In other words, the relation between syntax and formal semantic is clear in one direction (syntax $\rightarrow$ semantic), but not in the other direction (semantic $\rightarrow$ syntax). For example, assume the set of atomic propositions $\Omega = \{a, b\}$ and the sentence $\neg a \vee b$ then $M(\neg a \vee b) = \{\{\}, \{b\}, \{a, b\}\}$, however, assume $M(s) = \{\{b\}\}$ then s can be $\neg a \wedge b$, $\neg(\neg a \supset \neg b)$, etc. Furthermore, it is tedious to check for an arbitrary sentence s, if it is true in a particular interpretation δ or not, because we need to replace each proposition in s by its corresponding truth value in δ and following the interpretation rule of each connective to decide if s is true in δ or not. In addition, the number of possible interpretations is normally exponential in the number of atomic propositions ($2^{|\Omega|}$).

To overcome the previous shortcomings and to build another lattice, simpler and more convenient to IR than $\mathcal{B}_M$, we here propose an intermediate representation of logical sentences based on rewriting them in DNF. The nodes of the new lattice are flat sets of elements, and the partial order relation between nodes is equivalent to the validity of material implication.

We here use two mathematical frameworks, $\mathcal{PL}$ and lattices, which have similar notation. Therefore, and to eliminate vagueness, we differentiate between: 1- Lattice-related notation: *meet* ($\dot{\wedge}$), *join* ($\dot{\vee}$), *complement* ($\dot{\neg}$), and 2- Logic-related notation: *conjunction* ($\wedge$), *disjunction* ($\vee$), *negation* ($\neg$).

5.2.1 Intermediate Representation

The intermediate representation of logical sentences is essentially based on rewriting these sentences in their DNF form. The intermediate representation is another way to express logical sentences in $\mathcal{PL}$. It allows us to build a Boolean algebra $\mathcal{B}_\Theta$, where, the nodes of $\mathcal{B}_\Theta$ are simpler than the nodes of $\mathcal{B}_M$ and without loss of generality, because any logical sentence can be rewritten in DNF form, moreover, the partial order relation between the nodes of $\mathcal{B}_\Theta$ is equivalent to the validity of material implication.

This intermediate representation also facilitates the process of checking if an arbitrary sentence s is true or not in a particular interpretation δ, which is an essential process in logic-based IR models.

5.2.1.1 Logical Sentences in DNF

A logical sentence s in DNF is a disjunction of *clauses* and each clause is a conjunction of *literals* and each literal is either an atomic proposition or its negation. Assume the set Ω is a set of atomic propositions and it forms the alphabet of the logic (Definition C.1–P.185).

Definition 5.1 (Clauses Θ_Ω)**.** *A clause is a conjunction of literals, and each literal is an atomic proposition or its negation. Any clause s of the set of clauses Θ_Ω on the alphabet Ω is defined as follows:*

$$\forall s \in \Theta_\Omega, \exists \Omega_s \subseteq \Omega, s = \bigwedge_{a_i \in \Omega_s} b_i$$

where Ω_s is the set of atomic propositions that appear in s, and b_i is a literal. Any literal b_i is either an atomic proposition a_i or its negation $\neg a_i$. □

Definition 5.2 (DNF Sentences)**.** *A logical sentence written in DNF is a disjunction of clauses. Any sentence s written in DNF, using the set of clauses Θ_Ω, is defined as follows:*

$$\exists \Theta_s \subseteq \Theta_\Omega, s = \bigvee_{s_i \in \Theta_s} s_i$$

where Θ_s is the set of clauses that form s. □

5.2.1.2 The Intermediate Representation

For any clause $s \in \Theta_\Omega$, the set of propositions Ω_s that appear in s is splittable into two subsets: Ω_s^+ which contains the propositions occurring in their non-negative form, and Ω_s^- which contains the propositions occurring in their negative form, where $\Omega_s^+ \cup \Omega_s^- = \Omega_s$. We also define the set $\Omega_s^\pm = \Omega \setminus \Omega_s$, which contains the propositions that do not occur in s. Based on this splitting, we define our intermediate representation (Definition 5.3).

Definition 5.3 (Alphabet Splitting: The Intermediate Representation)**.** *Each clause $s \in \Theta_\Omega$ splits the alphabet Ω into three subsets of atomic propositions:*

- Ω_s^+ *contains the propositions* $a_i \in \Omega_s$ *where* $b_i = a_i$.
- Ω_s^- *contains the propositions* $a_i \in \Omega_s$ *where* $b_i = \neg a_i$.
- $\Omega_s^\pm = \Omega \setminus \Omega_s$ *contains the propositions that do not occur in* s.

where $\Omega_s^+ \cup \Omega_s^- = \Omega_s$. There are two special cases:

- *If ($\Omega_s^+ \cap \Omega_s^- \neq \emptyset$) then there is a proposition $a_i \in \Omega_s$ that appears in its negative and non-negative form within the same clause s, or in other words, $s = \cdots \wedge a_i \wedge \neg a_i \wedge \ldots$ and s is thus equivalent to the logical false F or it is **unsatisfiable**. For an unsatisfiable clause s, we put: $\Omega_s^+ = \Omega_s^- = \Omega$ and $\Omega_s^\pm = \emptyset$ because, the logical false F can be rewritten as $a_1 \wedge \neg a_1 \wedge \cdots \wedge a_n \wedge \neg a_n$.*
- *If ($\Omega_s = \emptyset$) then $\Omega_s^\pm = \Omega$, or equivalently, any atomic proposition can be either true or false in any model of s, and thus any interpretation is a possible model of s, which in turn means, s is **valid**. For a valid clause s, we put: $\Omega_s^+ = \Omega_s^- = \emptyset$ and $\Omega_s^\pm = \Omega$ because, whatever is the truth value of propositions, s will be true.*

We define a function μ that transforms each clause to sets of propositions, as follows:

$$\mu : \Theta_\Omega \to 2^\Omega \times 2^\Omega$$

where 2^Ω is the power set of Ω, and

$$\forall s \in \Theta_\Omega, \mu(s) = \begin{cases} (\emptyset, \emptyset) & \text{if} \quad \Omega_s = \emptyset \\ (\Omega, \Omega) & \text{if} \quad \Omega_s^+ \cap \Omega_s^- \neq \emptyset \\ (\Omega_s^+, \Omega_s^-) & \text{otherwise} \end{cases}$$

$\mu(s)$ is the intermediate representation of the clause s. We also define another function θ that transfers back any element $(x, y) \in 2^\Omega \times 2^\Omega$ to a clause in Θ_Ω, as follows:

$$\theta : 2^\Omega \times 2^\Omega \to \Theta_\Omega$$

where

$$\forall (x, y) \in 2^\Omega \times 2^\Omega, \theta(x, y) = \begin{cases} T & \text{if} \quad (x, y) = (\emptyset, \emptyset) \\ \left(\bigwedge_{a_i \in x} a_i\right) \wedge \left(\bigwedge_{a_j \in y} \neg a_j\right) & \text{otherwise} \end{cases}$$

where T is an abstract always-valid clause $\models T$, or in other words, T is equivalent to the logical true. □

Example 5.1. *Suppose $\Omega = \{a, b, c\}$, for the clause $s = \neg a \wedge b \in \Theta_\Omega$ we have: $\Omega_s = \{a, b\}$, $\Omega_s^+ = \{b\}$, $\Omega_s^- = \{a\}$, $\Omega_s^\pm = \{c\}$, and $\mu(s) = (\{b\}, \{a\})$. Assume $(\{a, b\}, \{c\}) \in 2^\Omega \times 2^\Omega$, then $\theta(\{a, b\}, \{c\}) = a \wedge b \wedge \neg c$.* □

Definition 5.4. *The intermediate representation of any logical sentence s written in DNF is the set of intermediate representations of its clauses Θ_s.* □

Back to the formal interpretation of an arbitrary logical sentence in $\mathcal{PL}$, and knowing that clauses are a special form of logical sentences, any clause $s \in \Theta_\Omega$ thus corresponds to a set of models $M(s)$, where $M(s) \models s$. In any interpretation δ, in order to be a model of the clause s, the propositions Ω_s^+ must be mapped to true, the propositions Ω_s^- must be mapped to false, and the propositions $\Omega_s^\pm$ can be mapped to true or false. Hence, the number of models $|M(s)|$ of any clause $s \in \Theta_\Omega$ is $|M(s)| = 2^{|\Omega_s^\pm|}$. Here, we consider the *Open World Assumption* (OWA).

In the following, we show the rules that are used to check if an arbitrary sentence, written in DNF, is true or not in an interpretation δ using our intermediate representation:

- If s is a clause $s \in \Theta_\Omega$ then $\{\delta\} \models s$ *iff*
  - the propositions of Ω_s^+ is mapped to true, or equivalently, $\Omega_s^+ \subseteq \delta$, **and**
  - the propositions of Ω_s^- is mapped to false, or equivalently, $\Omega_s^- \cap \delta = \emptyset$.
- For any sentence $s = s_1 \vee \cdots \vee s_i$, where $s_1, \ldots, s_i$ are clauses, $\{\delta\} \models s$ *iff* $\{\delta\} \models s_1$ **or** $\ldots$ **or** $\{\delta\} \models s_i$.

The checking if a logical sentence is true or not in an interpretation becomes a simple set-inclusion checking. Determining the truth of a logical sentence in an interpretation is an important and essential operation in logic-based IR models [Lalmas & Bruza, 1998; Nie, 1988; van Rijsbergen, 1986].

5.2.2 Intermediate Representation Based Boolean Algebra

Based on our proposed intermediate representation (Definition 5.3), another Boolean algebra $\mathcal{B}_\Theta$ (Theorem 5.1) is defined, which is different from the Boolean algebra $\mathcal{B}_M$ that is built based on the formal interpretation of $\mathcal{PL}$.

Theorem 5.1 (Intermediate Representation Based Boolean Algebra $\mathcal{B}_\Theta$)**.** *The algebraic structure* $\mathcal{B}_\Theta = (2^\Omega \times 2^\Omega, \wedge, \vee, \dot{\neg}, \top, \bot)$ *is a Boolean algebra, where:*

- 2^Ω *is the powerset of* Ω.
- ***meet*** *operation:* $\forall (x_1, y_1), (x_2, y_2) \in 2^\Omega \times 2^\Omega, (x_1, y_1) \wedge (x_2, y_2) = (x_1 \cap x_2, y_1 \cap y_2)$
- ***join*** *operation:* $\forall (x_1, y_1), (x_2, y_2) \in 2^\Omega \times 2^\Omega, (x_1, y_1) \vee (x_2, y_2) = (x_1 \cup x_2, y_1 \cup y_2)$
- ***complement*** *operation:* $\forall (x, y) \in 2^\Omega \times 2^\Omega, \dot{\neg}\,(x, y) = (\Omega \setminus x, \Omega \setminus y)$
- ***top*** *element:* $\top = (\Omega, \Omega)$
- ***bottom*** *element:* $\bot = (\emptyset, \emptyset)$

The partial order relation $\leq$ *defined on* $\mathcal{B}_\Theta$ *is:*

$$[(x_1, y_1) \leq (x_2, y_2)] \Leftrightarrow [(x_1 \subseteq x_2) \quad \textit{and} \quad (y_1 \subseteq y_2)]$$

Proof. The proof of this theorem can be directly established based on (Theorem B.1–P.179). □

Theorem 5.2. *The potential relationship between the material implication $\supset$ and the partial order relation $\leq$ defined on $\mathcal{B}_\Theta$ is:*

$$\forall s_1, s_2 \in \Theta_\Omega, [\models s_1 \supset s_2] \Leftrightarrow [\mu(s_2) \leq \mu(s_1)]$$

OR

$$\forall (x_1, y_1), (x_2, y_2) \in 2^\Omega \times 2^\Omega, [\mu(\theta(x_2, y_2)) \leq \mu(\theta(x_1, y_1))] \Leftrightarrow [\models \theta(x_1, y_1) \supset \theta(x_2, y_2)]$$

Proof. **Point 1.** Assume $\models s_1 \supset s_2$ then we have three possible cases:

- Unsatisfiable premises ($\not\models s_1$). The clause s_1 is unsatisfiable *iff* $\Omega^+_{s_1} \cap \Omega^-_{s_1} \neq \emptyset$. In this case, $\mu(s_1) = (\Omega, \Omega) = \top$ (Theorem 5.1), and then $\forall s_2 \in \Theta_\Omega, \mu(s_2) \leq \mu(s_1)$.
- Valid conclusions ($\models s_2$). The clause s_2 is valid *iff* $\Omega_{s_2} = \emptyset$. In this case, $\mu(s_2) = (\emptyset, \emptyset) = \bot$ (Theorem 5.1), and then $\forall s_1 \in \Theta_\Omega, \mu(s_2) \leq \mu(s_1)$.
- Otherwise: We know that $[\models s_1 \supset s_2] \Leftrightarrow [M(s_1) \subseteq M(s_2)]$, which means that every model of s_1 is also a model of s_2, or in other words, every proposition in s_2 must have the same truth value in both s_1 and s_2, and then $\Omega^+_{s_2} \subseteq \Omega^+_{s_1}$ and $\Omega^-_{s_2} \subseteq \Omega^-_{s_1}$, or equivalently, $\mu(s_2) \leq \mu(s_1)$.

Point 2. Assume $\mu(s_2) \leq \mu(s_1)$ then

$\mu(s_2) \leq \mu(s_1)$ means that $\Omega^+_{s_2} \subseteq \Omega^+_{s_1}$ and $\Omega^-_{s_2} \subseteq \Omega^-_{s_1}$, and then every proposition in s_2 has the same truth value in both s_1 and s_2, which means that every model of s_1 is also a model of s_2. In other words, $M(s_1) \subseteq M(s_2)$ and thus $\models s_1 \supset s_2$.

From point 1, we prove that:

$$[\models s_1 \supset s_2] \Rightarrow [\mu(s_2) \leq \mu(s_1)]$$

From point 2, we prove that:

$$[\models s_1 \supset s_2] \Leftarrow [\mu(s_2) \leq \mu(s_1)]$$

Similarly, we prove that

$$\forall (x_1, y_1), (x_2, y_2) \in 2^\Omega \times 2^\Omega, [\mu(\theta(x_2, y_2)) \leq \mu(\theta(x_1, y_1))] \Leftrightarrow [\models \theta(x_1, y_1) \supset \theta(x_2, y_2)]$$

□

Theorem 5.2 shows that the partial order relation $\leq$ defined on the Boolean algebra $\mathcal{B}_\Theta$ is equivalent to the validity of materiel implication $\supset$ between the clauses of Θ_Ω (Definition 5.1). However, this theorem talks about the validity of material implication between *clauses*. Later in this chapter, when describing our IR model, we will introduce the relation between the validity of material implication of two arbitrary logical sentences and the partial order relation defined on the Boolean algebra $\mathcal{B}_\Theta$.

Figure 5.1 shows the position of our proposed intermediate representation with respect to: the formal language (syntax) of $\mathcal{PL}$, the formal interpretation or semantic of $\mathcal{PL}$, and the models-based Boolean algebra $\mathcal{B}_M$. It also shows how to go from one world to another. The Boolean algebra $\mathcal{B}_\Theta$ is depicted as a component directly connected to our intermediate representation, and $\mathcal{B}_\Theta$ is simpler than $\mathcal{B}_M$ where the nodes are flat sets instead of sets of sets in $\mathcal{B}_M$,

Figure 5.1: The position of our intermediate representation.

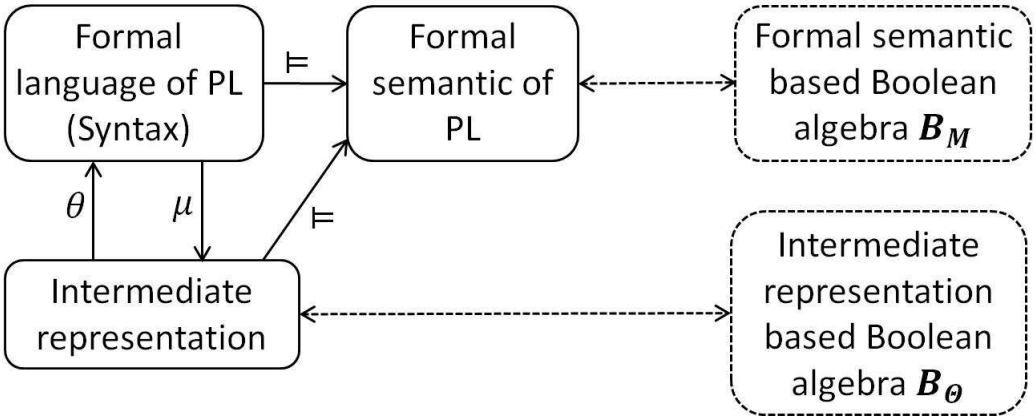


Table 5.1: The meaning of basic logical notions w.r.t. our intermediate representation.

Basic notions	$\mathcal{PL}$ formal language	Inter. representation	Boolean algebra $\mathcal{B}_\Theta$
A clause s	Conjunction of literals	$\mu(s) = (\Omega_s^+, \Omega_s^-)$	a node in $\mathcal{B}_\Theta$
A sentence s where $s = s_1 \vee s_2$	Disjunction of clauses s_1, s_2	$\{\mu(s_1), \mu(s_2)\}$	a set of nodes in $\mathcal{B}_\Theta$
Validity of material implication between clauses	$\vdash s_1 \supset s_2$ where s_1, s_2 are clauses	$\Omega_{s_2}^+ \subseteq \Omega_{s_1}^+$ $\Omega_{s_2}^- \subseteq \Omega_{s_1}^-$	$\mu(s_2) \leq \mu(s_1)$

and at the same time, the partial order relation defined on $\mathcal{B}_\Theta$ is equivalent to the validity of material implication between clauses Θ_Ω.

Furthermore, the Boolean algebra $\mathcal{B}_\Theta$ is built upon the *formal language* of $\mathcal{PL}$ through exploiting our proposed intermediate representation, whereas $\mathcal{B}_M$ is built upon the *formal semantic* of $\mathcal{PL}$. Actually, that helps to transform checking the validity of the material implication between two arbitrary clauses $\models s_1 \supset s_2$, which is a computationally intensive task, to simple set-inclusion checking $\mu(s_2) \leq \mu(s_1)$.

Table 5.1 reviews, within each mathematical world, the basic logical notions: a clause, a sentence written in DNF, and the validity of material implication between clauses.

5.3 Logic and Lattice Based IR Model

We proposed an intermediate representation for logical clauses of $\mathcal{PL}$ (Definition 5.3). We then built a Boolean algebra $\mathcal{B}_\Theta$ upon this representation (Theorem 5.1). In this section, we exploit this intermediate representation and the Boolean algebra $\mathcal{B}_\Theta$ in order to define an IR model. In general, to define an IR model, we mainly need to define four components: a document, a query, relevance (retrieval decision), degree of relevance (uncertainty).

5.3.1 Documents and Queries

In logic-based IR models, documents and queries are normally represented through logical sentences. In this study, documents and queries are DNF logical sentences. In other words, a document (or a query) is a set of clauses of Θ_Ω connected by disjunction. Since any logical sentence can be rewritten in the DNF form, there are thus no restrictions on the logical sentences that can be used to represent documents and queries, and hence there is no loss of generality. Furthermore, we consider the *Open World Assumption*. More formally, suppose we have a set of documents D and a query q:

Definition 5.5 (Document)**.** *Any document $d \in D$ is a DNF logical sentence, and it corresponds to one unique non-empty set of clauses $\Theta_d \subseteq \Theta_\Omega$ connected via disjunction, or equivalently:*

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

where $\Theta_d \neq \emptyset$. □

Definition 5.6 (Query)**.** *The query q is a DNF logical sentence, and it corresponds to one unique non-empty set of clauses $\Theta_q \subseteq \Theta_\Omega$ connected via disjunction, or equivalently:*

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

where $\Theta_q \neq \emptyset$. □

Definitions 5.5 & 5.6 represent documents and queries in the most general form. They exploit the full expressive power of $\mathcal{PL}$. Logic-based IR models, which are based on $\mathcal{PL}$, normally represent documents and queries (especially documents) as a conjunction of terms or as a clause, which is a special case of the previous two definitions. Furthermore, in our representation, we do not make any pre-assumption about the terms that do not appear in d, where, for simplicity, these terms are set to false in most models.

Representing documents and queries as disjunction of several clauses enables us to represent different views of them. Assume a document $d = s_1 \vee s_2$ where s_1 and s_2 are two clauses, then s_1 can represent the English content of d and s_2 the French content (multilingualism), or s_1 represents the textual content of d and s_2 the graphical content (multimodality), etc.

5.3.2 Relevance

Most studies in the logical IR define the relevance between a document d and a query q by a non-classical implication between them [Chevallet & Chiaramella, 1998; Chiaramella & Chevallet, 1992; Crestani & Lalmas, 2001; Meghini *et al.*, 1993; Nie, 1988; van Rijsbergen, 1986]; d is relevant to q *iff* d implies q, denoted $d \rightarrow q$. In other words, the retrieval decision is equivalent to check the *truth* of the non-classical implication $d \rightarrow q$.

We claim that the *truth* of the non-classical implication $d \rightarrow q$ can be replaced by the *validity* of the material implication $\models d \supset q$ (Hypothesis 4.1). More precisely, we represent the retrieval decision between d and q through the validity of material implication.

Relevance: d is relevant to q *iff* the material implication $d \supset q$ is valid, denoted $\models d \supset q$.

Assume that d is a document (Definition 5.5) and q is a query (Definition 5.6). Theorem 5.2 states that: if each of d and q corresponds to only one clause ($|\Theta_d| = |\Theta_q| = 1$) then $\models d \supset q$ is equivalent to $\mu(q) \leq \mu(d)$. Generally and according to the number of document's clauses $|\Theta_d|$ and query's clauses $|\Theta_q|$, we have the following cases:

Case 1. $|\Theta_d| = |\Theta_q| = 1$

The document and query are represented by only one clause. Suppose that $\Theta_d = \{s_d\}$ and $\Theta_q = \{s_q\}$ then,

$$[\models d \supset q] \Leftrightarrow [\mu(s_q) \leq \mu(s_d)] \tag{5.1}$$

see (Theorem 5.2).

Case 2. $|\Theta_d| = 1$ and $|\Theta_q| > 1$

The document is represented by only one clause, but the query is a disjunction of several clauses. We know that for any logical sentences s_1, s_2, s_3:

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

Then, suppose $\Theta_d = \{s_d\}$:

$$[\models d \supset q] \Leftrightarrow [\exists s_i \in \Theta_q, \mu(s_i) \leq \mu(s_d)] \tag{5.2}$$

This case represents the main assumption in most logic-based IR models, where d is a conjunctions of terms and q is any logical sentences of terms. Equation 5.2 has two main advantages:

- For a document d, it is sufficient to find a part of the query $s_i \in \Theta_q$ where $\models d \supset s_i$ for deciding that d is relevant to q (partial relevance).
- Instead of checking the validity of an implication, we check the set-inclusion between two sets of elements (very simple and implementable checking).

Case 3. $|\Theta_d| > 1$ and $|\Theta_q| = 1$

The document is a disjunction of several clauses, whereas the query is represented by only one clause. We know that for any logical sentences s_1, s_2, s_3:

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

Then, suppose $\Theta_q = \{s_q\}$:

$$[\models d \supset q] \Leftrightarrow [\forall s_i \in \Theta_d, \mu(s_q) \leq \mu(s_i)] \tag{5.3}$$

Case 4. $|\Theta_d| > 1$ and $|\Theta_q| > 1$ (the most general case)

Each of d and q is a disjunction of several clauses. We know that for any logical sentences s_1, s_2, s_3, s_4:

$$[(s_1 \vee s_2) \supset (s_3 \vee s_4)] \Leftrightarrow [[(s_1 \supset s_3) \vee (s_1 \supset s_4)] \wedge [(s_2 \supset s_3) \vee (s_2 \supset s_4)]]$$

Then,

$$[\models d \supset q] \Leftrightarrow [\forall s_i \in \Theta_d, \exists s_j \in \Theta_q, \mu(s_j) \leq \mu(s_i)] \tag{5.4}$$

This is the most general case where d and q can be any logical sentence, since any logical sentence can be rewritten in a DNF form.

We can see that using our intermediate representation, checking the validity of the material implication $\models d \supset q$ is transformed to a series of simple set-inclusion checking.

5.3.3 Uncertainty

It is known that IR is an uncertain process [Chiaramella & Chevallet, 1992]. Therefore, it is mandatory to define a measure for quantifying the validity of the implication $d \supset q$, written $U(\models d \supset q)$. It is rarely the case where $d \supset q$ is valid, so we need a measure to estimate the degree to which $d \supset q$ is valid, and then ranking documents according to the decreasing value of this measure.

According to (Definitions 5.5 & 5.6), documents and queries are sets of clauses, or equivalently, sets of nodes in $\mathcal{B}_\Theta$ (Definition 5.3), where $\mathcal{B}_\Theta$ is a Boolean algebra (Theorem 5.1). Knuth [Knuth, 2005] defines the Z function on lattices, where for any two distinct elements x and y of a lattice, $Z(x, y)$ measures the degree to which x includes or implies y. In other words, the Z function quantifies the partial order relation defined on a lattice.

First we must redefine the Z function on our Boolean algebra $\mathcal{B}_\Theta$. For any two nodes or elements $(x_1, y_1), (x_2, y_2) \in 2^\Omega \times 2^\Omega$ of the Boolean algebra $\mathcal{B}_\Theta$:

$$Z((x_1, y_1), (x_2, y_2)) = \begin{cases} 1 & \text{if} \quad (x_2, y_2) \leq (x_1, y_1) \\ 0 & \text{if} \quad (x_1, y_1) \wedge (x_2, y_2) = \bot \\ z & \text{otherwise, where} \quad 0 < z < 1 \end{cases} \tag{5.5}$$

The condition $(x_2, y_2) \leq (x_1, y_1)$ means that the material implication between the logical clauses that correspond to the two nodes $(x_1, y_1), (x_2, y_2)$ is valid (Theorem 5.2). The condition $(x_1, y_1) \wedge (x_2, y_2) = \bot$ means that $x_1 \cap x_2 = \emptyset$ and $y_1 \cap y_2 = \emptyset$, or equivalently, the logical clauses that correspond to the two nodes $(x_1, y_1), (x_2, y_2)$ use a very different set of propositions.

Let us now come back to our initial uncertain retrieval decision $U(\models d \supset q)$. In logic-based IR models, d and q are logical sentences. By rewriting d and q in DNF form, each of them becomes one or several clauses or equivalently nodes in $\mathcal{B}_\Theta$. Let us now assume that each of d and q is only one clause, in the next section we will discuss the general case. We propose to estimate the uncertainty $U(\models d \supset q)$ via the function Z (Equation 5.5), as follows:

$$U(\models d \supset q) = Z(\mu(d), \mu(q)) \tag{5.6}$$

We postulate that the previous rewriting (Equation 5.6) is reasonable, because:

- $Z(\mu(d), \mu(q)) = 1$ when $\mu(q) \leq \mu(d)$ which is equivalent to $\models d \supset q$ (Theorem 5.2). More precisely, when the implication $d \supset q$ is valid then $\mu(q) \leq \mu(d)$, and thus the value of Z will be equal to 1:

$$[Z(\mu(d), \mu(q)) = 1] \Leftrightarrow [\models d \supset q]$$

- $Z(\mu(d), \mu(q)) = 0$ when $(\Omega_d^+ \cap \Omega_q^+ = \emptyset)$ and $(\Omega_d^- \cap \Omega_q^- = \emptyset)$ which means that d and q use different propositions, or equivalently, d and q use different terms. By supposing that terms are independent then d and q are about very different things. For example, this case

correspond to the case when we try to match a query about '*dolphins*' with a document about '*grasslands*'.

More formally, the implication $d \supset q$ is unsatisfiable ($\not\models d \supset q$) when d is valid and q is unsatisfiable. According to (Definition 5.3), if d is valid then $\mu(d) = (\emptyset, \emptyset)$, and if q is unsatisfiable then $\mu(q) = (\Omega, \Omega)$. Therefore, if $\not\models d \supset q$ then $(\Omega_d^+ \cap \Omega_q^+ = \emptyset)$ and $(\Omega_d^- \cap \Omega_q^- = \emptyset)$. Accordingly, when the implication $d \supset q$ is unsatisfiable then Z will be equal to 0:

$$[\not\models d \supset q] \Rightarrow [Z(\mu(d), \mu(q)) = 0]$$

- $0 < Z(\mu(d), \mu(q)) < 1$ otherwise. This condition represents the case when $d \supset q$ is neither valid nor unsatisfiable. In IR, this condition corresponds to the case when there are some terms shared between d and q, which is the general case in IR.

If we assume that terms are independent then the main difference between $Z(\mu(d), \mu(q)) = 0$ and $0 < Z(\mu(d), \mu(q)) < 1$ is that in the former d and q use different terms, and they are thus about totally different subjects. Whereas in the latter there is some thing shared between d and q, or in other words, there is some thing that we can build upon it to establish the matching between d and q.

5.3.4 The Relevance Status Value RSV(d,q)

In this section, we generalize (Equation 5.6) from the case where each of d and q is only one clause to the case where d and q are any logical sentence. Our goal is to estimate the Relevance Status Value $RSV(d, q)$ between d and q.

Nie [Nie, 1988] differentiates between the two non-classical implications *Exhaustivity* $d \rightarrow q$ and *Specificity* $q \rightarrow d$. He proposes to write the matching score $RSV(d, q)$ as follows:

$$RSV(d, q) = F\left[U(d \rightarrow q), U(q \rightarrow d)\right] \tag{5.7}$$

According to (Hypothesis 4.1–P.74), we propose to check the validity of material implication instead of checking the truth of non-classical implication. Therefore, (Equation 5.7) becomes:

$$RSV(d, q) = F\left[U(\models d \supset q), U(\models q \supset d)\right] \tag{5.8}$$

We take this general form of matching score (Equation 5.8), and we build our discussion on it. According to the form of d and q, we have:

Case 1. $|\Theta_d| = |\Theta_q| = 1$

The document and query are represented by only one clause. Suppose that $\Theta_d = \{s_d\}$ and $\Theta_q = \{s_q\}$ then,

- $U(\models d \supset q) = Z(\mu(s_d), \mu(s_q))$ (Equation 5.6).
- $U(\models q \supset d) = Z(\mu(s_q), \mu(s_d))$.

$$\begin{aligned} RSV(d, q) &= F\left[U(\models d \supset q), U(\models q \supset d)\right] \\ &= F\left[Z(\mu(s_d), \mu(s_q)), Z(\mu(s_q), \mu(s_d))\right] \end{aligned} \tag{5.9}$$

Case 2. $|\Theta_d| = 1$ and $|\Theta_q| > 1$

The document is represented by only one clause, but the query is a disjunction of several clauses. Suppose that $\Theta_d = \{s_d\}$ then,

- $U(\models d \supset q) = G\left(Z(\mu(s_d), \mu(s_1)), Z(\mu(s_d), \mu(s_2)), \ldots, Z(\mu(s_d), \mu(s_n))\right)$ where $G : \mathbb{R}^n \to \mathbb{R}$, and $n = |\Theta_q|$. G must be consistent with (Equation 5.2). In order to build such type of function, G must be a *triangular conorm* function. To simplify the notation, we will refer to $G\left(Z(\mu(s_d), \mu(s_1)), \ldots, Z(\mu(s_d), \mu(s_n))\right)$ by $\underset{s_i \in \Theta_q}{G}\left(Z(\mu(s_d), \mu(s_i))\right)$.
- $U(\models q \supset d) = \underset{s_i \in \Theta_q}{G'}\left(Z(\mu(s_i), \mu(s_d))\right)$ where $G' : \mathbb{R}^n \to \mathbb{R}$ must be consistent with (Equation 5.3). In order to build such type of function, G' must be a *triangular norm* function.

The functions G and G' must satisfy the following conditions to be *triangular conorm* and *triangular norm*, respectively [Bělohlávek & Klir, 2011]: $\forall a, b, c \in [0, 1]$,

- Associativity: $G(a, G(b, c)) = G(G(a, b), c)$ and $G'(a, G'(b, c)) = G'(G'(a, b), c)$.
- Monotonicity: If $b \leq c$ then $G(a, b) \leq G(a, c)$ and $G'(a, b) \leq G'(a, c)$.
- Commutativity: $G(a, b) = G(b, a)$ and $G'(a, b) = G'(b, a)$.
- Boundary conditions: $G(a, 0) = a$ and $G'(a, 1) = a$.

For example, G can be the normal sum $+$ or the $\max$ function, and G' can be the normal product $\times$ or the $\min$ function.

$$\begin{aligned} RSV(d, q) &= F\left[U(\models d \supset q), U(\models q \supset d)\right] \\ &= F\left[\underset{s_i \in \Theta_q}{G}\left(Z(\mu(s_d), \mu(s_i))\right), \underset{s_i \in \Theta_q}{G'}\left(Z(\mu(s_i), \mu(s_d))\right)\right] \end{aligned} \tag{5.10}$$

Case 3. $|\Theta_d| > 1$ and $|\Theta_q| = 1$

The document is a disjunction of several clauses, whereas the query is represented by only one clause. Suppose that $\Theta_q = \{s_q\}$ then,

- $U(\models d \supset q) = \underset{s_i \in \Theta_d}{G'}\left(Z(\mu(s_i), \mu(s_q))\right)$
- $U(\models q \supset d) = \underset{s_i \in \Theta_d}{G}\left(Z(\mu(s_q), \mu(s_i))\right)$

$$\begin{aligned} RSV(d, q) &= F\left[U(\models d \supset q), U(\models q \supset d)\right] \\ &= F\left[\underset{s_i \in \Theta_d}{G'}\left(Z(\mu(s_i), \mu(s_q))\right), \underset{s_i \in \Theta_d}{G}\left(Z(\mu(s_q), \mu(s_i))\right)\right] \end{aligned} \tag{5.11}$$

Case 4. $|\Theta_d| > 1$ and $|\Theta_q| > 1$

Each of d and q is a disjunction of several clauses then,

- we define $G'' : 2^{\Theta_\Omega} \times 2^{\Theta_\Omega} \to \mathbb{R}$ where 2^{Θ_Ω} is the powerset of the set of clauses Θ_Ω, and

$$G''(\Theta_d, \Theta_q) = \underset{s_i \in \Theta_d}{G'}\left(\underset{s_j \in \Theta_q}{G}\left(Z(\mu(s_i), \mu(s_j))\right)\right) \tag{5.12}$$

- $U(\models d \supset q) = G''(\Theta_d, \Theta_q)$ (Equations 5.4 & 5.6).
- $U(\models q \supset d) = G''(\Theta_q, \Theta_d)$.

$$RSV(d,q) = F\left[U(\models d \supset q), U(\models q \supset d)\right] = F\left[G''(\Theta_d, \Theta_q), G''(\Theta_q, \Theta_d)\right] \tag{5.13}$$

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

5.4 Discussion

In the previous section, we presented a new IR model, starting from the definition of documents and queries, passing through representing the retrieval decision and its uncertainty, and ending with computing the matching score $RSV(d,q)$. In this section, we discuss the main theoretical properties of our proposed model, and we explore its main capabilities, which include the ability to formalize the van Rijsbergen assumption about replacing $U(d \rightarrow q)$ by $P(q|d)$, rewriting Exhaustivity and Specificity in an easy to implement form, and the model is also capable of reproducing most of classical IR models.

5.4.1 Formalizing Van Rijsbergen's Assumption

Van Rijsbergen [van Rijsbergen, 1986] assumes, in an intuitive manner, that the uncertainty of the retrieval decision $U(d \rightarrow q)$ between a document d and a query q can be estimated through the conditional probability $P(q|d)$. Although this assumption is well accepted in IR community, but to our knowledge it is not yet well-formalized. For formalizing it, we consider the Boolean algebra $\mathcal{B}_M$, which is built upon the formal interpretation of $\mathcal{PL}$ (Theorem C.2–P.188). We also replace the truth of the non-classical implication $d \rightarrow q$ by the validity of the material implication $\models d \supset q$ (Hypothesis 4.1).

If d and q are two arbitrary logical sentences then they correspond to two sets of models $M(d)$ and $M(q)$, or equivalently two nodes in the Boolean algebra $\mathcal{B}_M$.

First, we redefine the Z function on the Boolean algebra $\mathcal{B}_M$ as follows: for any two sets of interpretations $x, y \in 2^{2^\Omega}$,

$$Z(x,y) = \begin{cases} 1 & \text{if} \quad y \subseteq x \\ 0 & \text{if} \quad x \cap y = \emptyset \\ z & \text{otherwise, where} \quad 0 < z < 1 \end{cases} \tag{5.14}$$

On the one hand, we propose to estimate $U(\models d \supset q)$ via $Z(M(q), M(d))$:

$$U(\models d \supset q) = Z(M(q), M(d)) \tag{5.15}$$

because,

- $Z(M(q), M(d)) = 1$ when $M(d) \subseteq M(q)$ which is equivalent to $\models d \supset q$ (Theorem C.1–P.188). More precisely, when the implication $d \supset q$ is valid then $M(d) \subseteq M(q)$, and thus the value of Z will be equal to 1:

$$[Z(M(q), M(d)) = 1] \Leftrightarrow [\models d \supset q]$$

- $Z(M(q), M(d)) = 0$ when $M(d) \cap M(q) = \emptyset$ which means that if d is true then q is false and vice-versa. Therefore, the implication $d \supset q$ can be true or false but it is impossible to be valid (Theorem C.3–P.188).
- $0 < Z(M(q), M(d)) < 1$ otherwise. In this case, even the implication $d \supset q$ is not explicitly valid, it is possible to find at least a subset of models, e.g. $M(d \wedge q) = M(d) \cap M(q)$, in which $d \supset q$ becomes valid.

Assuming that Z is consistent with all structural properties of the Boolean algebra $\mathcal{B}_M$, then Z is the conditional probability (Equation B.8–P.178). Therefore,

$$U(\models d \supset q) = Z(M(q), M(d)) = P(M(q)|M(d))$$

On the other hand, we know that each node in $\mathcal{B}_M$ represents a set of models of a set of logically equivalent 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,

$$U(d \rightarrow q) = U(\models d \supset q) = Z(M(q), M(d)) = P(M(q)|M(d)) = P(q|d) \tag{5.16}$$

Equation 5.16 formalizes the definition of $U(d \rightarrow q)$ that is presented by van Rijsbergen [van Rijsbergen, 1986]. To our knowledge, this is the first study that present a mathematical formalization for the Rijsbergen's assumption.

5.4.2 Exhaustivity & Specificity

We reconsider the Boolean algebra $\mathcal{B}_\Theta$ which is built based on our intermediate representation. We take the definition of the degree of implication function Z that is presented in (Equation 5.5). If Z is consistent with the structure of the Boolean algebra $\mathcal{B}_\Theta$ then Z satisfies: the *Sum* rule (Equation B.4–P.178), the *First Product* rule (Equation B.5–P.178), the *Second Product* rule (Equation B.6–P.178), and the *Bayes' Theorem* rule (Equation B.7–P.178). By considering these rules, it is possible to draw some valuable and useful conclusions.

We suppose that d is one single clause, or equivalently one node $\mu(d)$ in $\mathcal{B}_\Theta$, and the same for q. The conclusions that we draw here can be generalized to the cases where d and q are any logical sentences.

Here, we consider the definitions of Exhaustivity and Specificity that are introduced by Nie [Nie, 1988]. Nie represents Exhaustivity through $U(d \rightarrow q)$ and Specificity through $U(q \rightarrow d)$. According to (Hypothesis 4.1–P.74) and (Equation 5.6), Exhaustivity becomes $Z(\mu(d), \mu(q))$ and Specificity becomes $Z(\mu(q), \mu(d))$.

According to the *Sum* rule of Z:

$$Z(\mu(d) \wedge \mu(q), \mu(q)) = Z(\mu(d), \mu(q)) + Z(\mu(q), \mu(q)) - Z(\mu(d) \vee \mu(q), \mu(q))$$

but we know that $Z(\mu(q), \mu(q)) = 1$ because $\mu(q) \leq \mu(q)$, and also $Z(\mu(d) \vee \mu(q), \mu(q)) = 1$ because $\mu(q) \leq \mu(d) \vee \mu(q)$. Hence,

$$U(d \rightarrow q) = U(\models d \supset q) = Z(\mu(d), \mu(q)) = Z(\mu(d) \wedge \mu(q), \mu(q)) \tag{5.17}$$

In the same way, according to the *Sum* rule of Z:

$$Z(\mu(d) \curlywedge \mu(q), \mu(d)) = Z(\mu(d), \mu(d)) + Z(\mu(q), \mu(d)) - Z(\mu(d) \curlyvee \mu(q), \mu(d))$$

but we know that $Z(\mu(d), \mu(d)) = 1$ because $\mu(d) \leq \mu(d)$, and also $Z(\mu(d) \curlyvee \mu(q), \mu(d)) = 1$ because $\mu(d) \leq \mu(d) \curlyvee \mu(q)$. Hence,

$$U(q \rightarrow d) = U(\models q \supset d) = Z(\mu(q), \mu(d)) = Z(\mu(d) \curlywedge \mu(q), \mu(d)) \tag{5.18}$$

The main difference, between the new definitions of Exhaustivity and Specificity (Equations 5.17 & 5.18) and the original definitions, is that instead of comparing two different objects d and q for estimating Exhaustivity and Specificity, we here compare d (or q) with a part of it $\mu(d) \curlywedge \mu(q)$. Using *Bayes' Theorem* rule, we obtain:

$$Z(\mu(d), \mu(q)) = \frac{Z(\mu(d), \top)}{Z(\mu(q), \top)} \times Z(\mu(q), \mu(d)) \tag{5.19}$$

where $Z(x, \top)$ represents prior probability, and can be arbitrary assigned [Knuth, 2005]. Equation 5.19 clarifies the relation between Exhaustivity and Specificity, where we can see that Exhaustivity is monotonically increasing with respect to the size of documents and monotonically decreasing with respect to the size of queries, whereas the case is reversed with Specificity.

5.4.3 General Framework

The proposed model in this thesis forms a general IR framework and most classical IR models can be derived from it. Any instance $\mathfrak{I}$ of our model is determined through providing a precise definition of the following elements:

$$\mathfrak{I} = (\mu, F, Z, G, G') \tag{5.20}$$

- The function $\mu : \Theta_\Omega \rightarrow 2^\Omega \times 2^\Omega$ (Definition 5.3), is a function to translate logical clauses into sets of atomic propositions.
- Providing a precise definition of the components of (Equation 5.13):
  - $F : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$ is a function to merge two numerical values.
  - Z is the degree of inclusion or implication function defined on $\mathcal{B}_\Theta$. If Z is consistent with the structure of $\mathcal{B}_\Theta$ then it corresponds to a probability function P. However, Z is not forcibly consistent with the whole structure of the Boolean algebra, so in general: $Z : (2^\Omega \times 2^\Omega) \times (2^\Omega \times 2^\Omega) \rightarrow \mathbb{R}$. The only restriction with respect to Z, if we want to exploit the two results (Equations 5.17 & 5.18), is that the Z must satisfy the *Sum* rule.
  - $G : \mathbb{R}^m \rightarrow \mathbb{R}$ is a function to merge several numerical values, and it must be triangular conorm.
  - $G' : \mathbb{R}^m \rightarrow \mathbb{R}$ is a function to merge several numerical values, and it must be triangular norm.

5.4.3.1 Boolean Model (BM)

Boolean model assumes that the document d is a clause and the query q is any logical sentence. The document is relevant to the query *iff* the implication $d \supset q$ is valid. As we see the Boolean model corresponds to the direct application of our model.

Assume that each proposition $a_i \in \Omega$ corresponds to one term in the document collection D. A document $d \in D$ is written as follows:

$$d = \bigwedge_{a_i \in \Omega} b_i$$

where $b_i = a_i$ if the term a_i indexes d, or $b_i = \neg a_i$ otherwise. In other words, d is a conjunction of the terms that describe it and the conjunction of the negative form of the other terms (Close World Assumption). The query q is any logical sentence, and thus it corresponds to a set of clauses Θ_q.

The retrieval decision in the Boolean model is binary, and thus we do not need the degree of inclusion function Z. Therefore, to decide if d is relevant to q or not, it is sufficient to check the condition in (Equation 5.2). The instance $\mathfrak{I}_{BM}$ is:

- The function μ is the same as in (Definition 5.3), but in Boolean model the document is always one clause.
- We do not need to define the functions F, Z, G, G' because the retrieval decision in the Boolean model is binary, and thus it is sufficient to check the condition in (Equation 5.2).

5.4.3.2 Language Models (LM)

Language models estimate the similarity between a document d and a query q through assuming that d is a language, represented as a probability distribution, and q is a phrase, and then measuring the ability of the language d to reproduce the phrase q, denoted $P(q|\pi_d)$, where π_d is the probability distribution defined on d [Ponte & Croft, 1998].

Assume that each proposition $a_i \in \Omega$ corresponds to one term in the document collection D. A document $d \in D$ is written as follows:

$$d = \bigwedge_{a_i \in \Omega_d} a_i \tag{5.21}$$

where $\Omega_d \neq \emptyset$ is the set of terms that index or describe d, and a_i is a term indexing d, or equivalently, d is *about* a_i. In other words, d is a conjunction of the terms that describe it. For any document d, we have: $\Omega_d^- = \emptyset$, or in other words, the negation of terms is not modeled. The query q is represented in the same way,

$$q = \bigwedge_{a_i \in \Omega_q} a_i \tag{5.22}$$

Hence, $d, q \in \Theta_\Omega$ are two clauses, and they correspond to two distinct nodes $\mu(d) = (\Omega_d, \emptyset)$ and $\mu(q) = (\Omega_q, \emptyset)$, respectively. Therefore, $G''(\{d\}, \{q\}) = Z(\mu(d), \mu(q))$ and $G''(\{q\}, \{d\}) =$

$Z(\mu(q), \mu(d))$ (Equation 5.12). We choose F as the weighted-sum between two values, so (Equation 5.13) can be rewritten as follows:

$$RSV(d, q) = \alpha \times Z(\mu(d), \mu(q)) + \beta \times Z(\mu(q), \mu(d))$$

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

$$RSV(d, q) = Z(\mu(q), \mu(d))$$

Assume Z is consistent with all structural properties of the Boolean algebra $\mathcal{B}_\Theta$, then

$$RSV(d, q) = Z(\mu(q), \mu(d)) = P(\mu(q)|\mu(d))$$

We know that $\mu(d) = (\Omega_d, \emptyset)$ and $\mu(q) = (\Omega_q, \emptyset)$, then it is possible to rewrite the previous equation as follows:

$$RSV(d, q) = P(\Omega_q|\Omega_d)$$

Now, suppose that the elements of Ω_q are conditionally independent, and let us define a probability distribution π_d on the set Ω_d then

$$RSV(d, q) = \prod_{a_i \in \Omega_q} P(a_i|\pi_d)$$

which is the general form of language models. Therefore, language models are instances of our general framework. The instance $\mathfrak{I}_{LM}$ is:

- The function μ is the same as in (Definition 5.3), but language models do not take the negative terms into account. Therefore, for any clause s the set of negative propositions Ω_s^- is empty.
- The function F is the weighted sum.
- The function Z is a probability function.
- The functions G and G' are the identity function, where $\forall x \in \mathbb{R}, G(x) = G'(x) = x$.

5.4.3.3 Probabilistic Models (PM)

In probabilistic models, each query q determines two classes of documents: Relevant (R) and Non-Relevant (NR). Any new document d is ranked by comparing the probability that d belongs to R with the probability that d belongs to NR. Probabilistic models depend on the *Probability Ranking Principle* [Robertson, 1977], 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:

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

R, NR are binary random variables, and 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 Bayes' rule and with some simplifications, the previous formula becomes:

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

To estimate the two probabilities, we should have samples of the relevant and non-relevant documents of each query. Some studies [Hiemstra & de Vries, 2000; Lavrenko & Croft, 2001; Zhai, 2008] claim that R and NR can be seen as two sets of relevant and non-relevant documents for a specific query, respectively. Robertson et al. [Robertson & Jones, 1976] also assume that a set of relevance judgments 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 $\mathcal{B}_\Theta$, we obtain: Assume that each proposition $a_i \in \Omega$ corresponds to one term in the document collection D,

- Any document $d \in D$ has the same definition presented in LMs (Equation 5.21), so it corresponds to only one node $\mu(d) = (\Omega_d, \emptyset)$ in $\mathcal{B}_\Theta$.
- Any query q has the same definition presented in LMs (Equation 5.22), so it corresponds to only one node $\mu(q) = (\Omega_q, \emptyset)$ in $\mathcal{B}_\Theta$.
- We know that any document d_i in R is relevant to q and satisfies $\models d_i \supset q$. By assuming that R is a disjunction of a set of documents $R = d_1 \vee \cdots \vee d_k$ then $\models R \supset q$.
- We know that any document d_i in NR is non-relevant to q and satisfies $\not\models d_i \wedge q$[1]. By assuming that NR is also a disjunction of a set of documents $NR = d_1 \vee \cdots \vee d_l$ then $\not\models NR \wedge q$.
- The retrieval decision can be reformulated as follows: 'd **is relevant to** q ***if*** $\models d \supset R$ **and** $\not\models d \supset NR$' (Theorem 5.3). By taking the degree of implication Z into account (Equation 5.5), we have:

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

  $G''(\{d\}, R)$ can be simplified to $\underset{d_i \in R}{G}(Z(\mu(d), \mu(d_i)))$. Since $\mu(d) = (\Omega_d, \emptyset)$ and $\mu(d_i) = (\Omega_{d_i}, \emptyset)$ we replace $Z(\mu(d), \mu(d_i))$ by $Z(\Omega_d, \Omega_{d_i})$. We choose the $\max$ function to replace G. In addition, assume that Z is consistent with all structural properties of the Boolean algebra $\mathcal{B}_\Theta$, then it corresponds to a conditional probability. We also suppose that the elements of Ω_d are conditionally independent. Finally, we obtain the following ranking formula:

  $$RSV(d,q) \propto \prod_{a_i \in \Omega_d} \frac{P(a_i|\Omega_R)}{P(a_i|\Omega_{NR})}$$

  where $(\Omega_R, \emptyset)$ is the node $\mu(d_i)$ that maximizes $\underset{d_i \in R}{G}(Z(\mu(d), \mu(d_i)))$, whereas (Ω_{NR}, ϕ) is the node $\mu(d_i)$ that maximizes $\underset{d_i \in NR}{G}(Z(\mu(d), \mu(d_i)))$.

[1] $\not\models d_i \wedge q$ means that $d_i \wedge q$ is false in all interpretations, or equivalently, there is no an interpretation validating both d_i and q.

The previous formula is the general form of probabilistic models. Therefore, probabilistic models are instances of our general framework. The instance $\mathfrak{I}_{PM}$ is:

- The function μ is the same as in (Definition 5.3), but probabilistic models do not take the negative terms into account. Therefore, for any clause s the set of negative propositions Ω_s^- is empty.
- The function F is the weighted sum.
- The function Z is a probability function.
- The function G is the $\max$ function.
- The function G' is the identity function, where $\forall x \in \mathbb{R}, G'(x) = x$.

Theorem 5.3. *In probabilistic models, a document* d *is relevant to a query* q *if:*

$$\models d \supset R \quad \textit{and} \quad \not\models d \supset NR$$

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

Case 1. If $\not\models d \supset R$:
$\not\models d \supset R$ means that,
$\not\models (d \supset d_1^R) \vee \cdots \vee (d \supset d_k^R)$,
$\forall 1 \leq i \leq k, \not\models d \supset d_i^R$,
We know that $\models d_i^R \supset q$, and we also know from (Theorem C.3) that if $d \supset d_i^R$ is unsatisfiable and $d_i^R \supset q$ is valid then $d \supset q$ is not valid. That means d is not relevant to q in this case.

Case 2. If $\models d \supset R$ and $\models d \supset NR$:
$\models d \supset NR$ means that,
$\models (d \supset d_1^{NR}) \vee \cdots \vee (d \supset d_l^{NR})$,
$\exists 1 \leq i \leq l, \models d \supset d_i^{NR}$,
$\models \neg d \vee d_i^{NR}$,
$\not\models \neg(\neg d \vee d_i^{NR})$,
we know that $\not\models d_i^{NR} \wedge q$ then $\not\models \neg(\neg d \vee d_i^{NR}) \vee (d_i^{NR} \wedge q)$,
$\not\models \big(\big(d \wedge \neg d_i^{NR}\big) \vee d_i^{NR}\big) \wedge \big(\big(d \wedge \neg d_i^{NR}\big) \vee q\big)$,
$\not\models (d \vee q) \wedge \big(\big(d \vee d_i^{NR}\big) \wedge (\neg d_i^{NR} \vee q)\big)$, we know that $\big(d \vee d_i^{NR}\big) \wedge (\neg d_i^{NR} \vee q)$ can be rewritten as $(d \wedge q) \vee (d \wedge \neg d_i^{NR}) \vee (d_i^{NR} \wedge q)$, we also know that $\not\models \neg(\neg d \vee d_i^{NR})$ and $\not\models (d_i^{NR} \wedge q)$ then,
$\not\models (d \vee q) \wedge (d \wedge q)$,
$\not\models (d \wedge q)$,
We know from (Theorem C.3) that if $d \wedge q$ is unsatisfiable then $d \supset q$ is not valid. That means d is not relevant to q in this case.

We must thus change our hypotheses in Case 1 & Case 2. □

Furthermore, 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 $\uparrow x$ of a node x in our lattice $\mathcal{B}_\Theta$:

$$\forall x \in 2^\Omega \times 2^\Omega, \uparrow x = \{x' | x' \in 2^\Omega \times 2^\Omega, x \leq x'\}$$

We define the non-relevant documents NR:

$$NR = \{\theta(x) | x \in 2^{\Omega} \times 2^{\Omega}, Z(x, \mu(q)) = 0\}$$

Now, it is possible to define the set of relevant documents R,

$$R = \{\theta(x) | x \in \uparrow \mu(q)\} \setminus NR$$

where $\forall d_i \in R, \mu(q) \leq \mu(d_i)$ which means that $d_i \supset q$ is valid which in its turn means that d_i is relevant to q.

The lattice $\mathcal{B}_{\Theta}$ allows us to define the family of probabilistic models. Moreover, it allows us to determine the relevant and non-relevant documents in advance, which is very important to estimate the two probabilities $P(d|R, q)$ and $P(d|NR, q)$.

5.4.3.4 Vector Space Model (VSM).

Vector space models assume that both documents and queries are vectors in the same term space. According to VSMs, the similarity between a document d and a query q is either the inverse of the Euclidean distance between them or the cosine of the angle between them [Salton *et al.*, 1975].

Assume that each proposition $a_i \in \Omega$ corresponds to one term in the document collection D. Suppose that any document $d \in D$ has the same definition presented in LM (Equation 5.21), so d corresponds to only one node $\mu(d) = (\Omega_d, \emptyset)$ in $\mathcal{B}_{\Theta}$. Suppose that any query q has the same definition presented in LM (Equation 5.22), so q corresponds to only one node $\mu(q) = (\Omega_q, \emptyset)$ in $\mathcal{B}_{\Theta}$.

For any node $x = (x^+, x^-) \in 2^{\Omega} \times 2^{\Omega}$ in $\mathcal{B}_{\Theta}$, we build a binary vector $\overrightarrow{x}$ as follows:

$$\overrightarrow{x} = \langle w_1, \ldots, w_n \rangle \tag{5.23}$$

where $w_i = 1$ if $a_i \in x^+$, or $w_i = 0$ otherwise. We define the following two operations:

Production $\forall x, y \in 2^{\Omega} \times 2^{\Omega}$,

$$\overrightarrow{x} \otimes \overrightarrow{y} = \langle w_1^x \times w_1^y, \ldots, w_n^x \times w_n^y \rangle$$

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

$$\overrightarrow{x \wedge y} = \overrightarrow{x} \otimes \overrightarrow{y} \tag{5.24}$$

Addition $\forall x, y \in 2^{\Omega} \times 2^{\Omega}$,

$$\overrightarrow{x} \oplus \overrightarrow{y} = \langle w_1^x + w_1^y - w_1^x \times w_1^y, \ldots, w_n^x + w_n^y - w_n^x \times w_n^y \rangle$$

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

$$\overrightarrow{x \vee y} = \overrightarrow{x} \oplus \overrightarrow{y} \tag{5.25}$$

Let us define the Z function as the inner-product $(\cdot)$ between the vectors of two nodes, as follows:

$$Z(x, y) = \frac{\overrightarrow{x} \cdot \overrightarrow{y}}{|\overrightarrow{y}|} \tag{5.26}$$

where $|\overrightarrow{y}| = \sum_{a_i} w_i$. Note that, the previous definition of Z satisfies the sum rule, where:

$$\frac{(\overrightarrow{x} \oplus \overrightarrow{y}) \cdot \overrightarrow{t}}{|\overrightarrow{t}|} = \frac{\overrightarrow{x} \cdot \overrightarrow{t}}{|\overrightarrow{t}|} + \frac{\overrightarrow{y} \cdot \overrightarrow{t}}{|\overrightarrow{t}|} - \frac{(\overrightarrow{x} \otimes \overrightarrow{y}) \cdot \overrightarrow{t}}{|\overrightarrow{t}|} \tag{5.27}$$

We choose F as the weighted-sum between two values, so (Equation 5.13) becomes:

$$RSV(d, q) = \alpha \times Z(\mu(d), \mu(q)) + \beta \times Z(\mu(q), \mu(d))$$

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

$$RSV(d, q) = Z(\mu(d), \mu(q))$$

Replacing Z in the previous equation by the definition presented in (Equation 5.26), then:

$$RSV(d, q) = \frac{\overrightarrow{d} \cdot \overrightarrow{q}}{|\overrightarrow{q}|}$$

$|\overrightarrow{q}|$ will not affect the ranking. Finally we obtain,

$$RSV(d, q) = \overrightarrow{d} \cdot \overrightarrow{q}$$

This is one of the different forms of VSMs (remember that Z is the inner-product function). The instance $\mathfrak{I}_{VSM}$ is:

- The function μ is the same as in (Definition 5.3), but vector space models do not take the negative terms into account. Therefore, for any clause s the set of negative propositions Ω_s^- is empty.
- The function F is the weighted-sum.
- The function Z is the inner-product between two vectors.
- The functions G and G' are the identity functions, where $\forall x \in \mathbb{R}, G(x) = G'(x) = x$.

5.4.3.5 Inference Networks (IN)

Inference network models estimate the similarity between documents and queries through initiating the network by an interior probability (the chosen document), and then updating the probabilities of the other nodes until getting the posterior probability of the intended node (the query node) [Turtle & Croft, 1990, 1991].

Assume that each proposition $a_i \in \Omega$ corresponds to one term in the document collection D. Suppose that any document $d \in D$ has the same definition presented in LM (Equation 5.21), so d corresponds to only one node $\mu(d) = (\Omega_d, \emptyset)$ in $\mathcal{B}_\Theta$. Suppose that each term a_i in the query q is represented by a node $\mu(a_i) = (\{a_i\}, \emptyset)$. Note that, the document is a logical clause $d \in \Theta_\Omega$

and the query terms are propositions, it is thus possible to represent them using binary random variables, where the binary random variable $A_i = 1$ if a_i appears in its positive form in q, or $A_i = 0$ otherwise.

Hence, d corresponds to one node whereas q corresponds to a set of nodes (a node for each term a_i). Therefore, $G''(\{d\}, q) = \underset{a_i \in q}{G} (Z(\mu(d), \mu(a_i)))$ and $G''(q, \{d\}) = \underset{a_i \in q}{G'} (Z(\mu(a_i), \mu(d)))$. We choose F as the weighted-sum between two values, so (Equation 5.13) is rewritten as follows:

$$RSV(d, q) = \alpha \times \underset{a_i \in q}{G} (Z(\mu(d), \mu(a_i))) + \beta \times \underset{a_i \in q}{G'} (Z(\mu(a_i), \mu(d)))$$

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

$$RSV(d, q) = \underset{a_i \in q}{G'} (Z(\mu(a_i), \mu(d)))$$

We know that $\mu(a_i) = (\{a_i\}, \emptyset)$ and $\mu(d) = (\Omega_d, \emptyset)$. We also know that $\mathcal{B}_\Theta$ is a Boolean algebra. Therefore, we replace $Z(\mu(a_i), \mu(d))$ by $P(A_i|\Omega_d)$. Moreover, assume that $bel_d(A_i) = P(A_i|\Omega_d)$ if $a_i \in \Omega_d$ or 0 otherwise, then,

$$RSV(d, q) = \underset{a_i \in q}{G'} (bel_d(A_i))$$

which is the general form of Inference Network models. The instance $\Im_{IN}$ is:

- The function μ is the same as in (Definition 5.3). However, d is only one node in $\mathcal{B}_\Theta$ whereas q is represented by several nodes, one for each term.
- The function F is the weighted sum.
- The function Z is a probability function.
- According to the implicit relation between query terms [Metzler & Croft, 2004], e.g. AND, OR, NOT, etc., we can build the appropriate form of G' for reproducing the standard query operators (#MAX, #AND, #OR, #NOT, #SUM, #WSUM) [Metzler & Croft, 2004]. In addition, it is possible to modelize the notion that any information need corresponds to a set of queries.

5.5 Conclusion

We present in this chapter a new theoretical framework for representing documents, queries, and the retrieval decision including a ranking mechanism. We use the Propositional Logic ($\mathcal{PL}$), as underlying logical framework, for representing documents and queries, and then we claim that the retrieval decision corresponds to the validity of material implication between a document and a query, denoted $\models d \supset q$. Furthermore, we propose an intermediate representation to logical sentences (Definition 5.3), where after rewriting any logical sentence s in its DNF form, we transform each clause of it to two sets of atomic propositions: one containing the propositions that must be true in any model of this clause of s, and another containing the propositions that must be false in any model. The intermediate representation enables us to transform checking the validity of material implication $\models d \supset q$ from a difficult and formal-interpretation based checking to a series of simple set-inclusion checking. After that, we position the intermediate

representations of documents and queries on a lattice, more precisely on a Boolean algebra. We finally exploit the degree of implication metric Z, defined on lattices, for representing the ranking mechanism.

On the one hand, this model presents a new vision of logic-based IR models through exploiting the implicit link between lattices and $\mathcal{PL}$. On the other hand, it presents a general IR framework capable of representing the classical IR models, like language models, probabilistic models, vector space models, and inference networks.

Another 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 (Equations B.4 & B.5 & B.6 & B.7).

The IR model of Losada et al. [Losada & Barreiro, 2001] is supposed to be the closest, among logic-based IR models, to our model. However, the main difference between their model and our model is that they use Belief Revision (BR) to estimate uncertainty, whereas, we use the potential relation between $\mathcal{PL}$ and lattices. Moreover, they are forced to do some ad-hoc simplifications to make their model computationally feasible, whereas, this is not the case in our model. Our model is also more general because it is capable of representing most classical IR models. In addition, the usage of lattice theory to estimate uncertainty allows us to deduce some interesting conclusions concerning the assumption of van Rijsbergen about estimating $U(d \rightarrow q)$ via $P(q|d)$, and also concerning the two theoretical notions: Exhaustivity and Specificity.

The main advantages of our model can be reviewed in the following points:

- The retrieval decision is the classical material implication $\models d \supset q$. The connective '$\rightarrow$', in the IR logical implication $d \rightarrow q$, is thus a part of the formal language of $\mathcal{PL}$. Actually, this is a very important property, because with this property it is possible to build an IR model based on the formal language of $\mathcal{PL}$ instead of its formal interpretation, where the formal interpretation of $\mathcal{PL}$ contains $2^{|\Omega|}$ different interpretations, which is a very huge number knowing that $|\Omega|$ is the number of atomic propositions. In IR, $|\Omega|$ could arrive to tens of thousands.
- Different from Losada et al., our model is based on clauses comparison instead of formal interpretation comparison, and without any type of ad-hoc simplifications. Therefore, the computation time of our algorithms is visible, and it is rather easy to build operational models of our theoretical model, as we will see in (Chapter 6–P.101).
- The uncertainty $U(d \rightarrow q)$ is estimated using lattice theory. On the one hand, the formal definition of U exactly corresponds to the degree to which d implies q, and it is not defined in an ad-hoc way like other models. On the other hand, positioning documents and queries on a lattice, representing the retrieval decision as a partial order relation, and estimating uncertainty through quantifying the partial order relation, all that lead to a very flexible framework susceptible to be in future developed in several ways.
- Our model provides a mathematical formalization of the very early van Rijsbergen's intuitional assumption about estimating the uncertainty $U(d \rightarrow q)$ through the conditional probability $P(q|d)$.
- Our model provides an implementable version of the very abstract notions Exhaustivity and Specificity, which are introduced by Nie [Nie, 1988]. It rewrites Exhaustivity and

Specificity, in a way that instead of comparing two different objects d and q, it compares d (or q) with a part of it $d \wedge q$.

- Our model is general enough to reproduce most classical IR models.
- Our model is *implicitly* capable of integrating an external knowledge into the retrieval process through the possibility to build several document and query representations.

The main disadvantage of our model (in its current version) is that it is not yet capable of *explicitly* integrating an external knowledge into the IR model, although, it is possible to do that in an implicit way. However, one of our main perspectives is to extend our model to be able to explicitly integrate an external knowledge.

Chapter 6

Instances of our Model

6.1 Introduction

In (Chapter 5–P.77), we presented a theoretical Information Retrieval (IR) model through proposing an intermediate representation for the logical sentences in Propositional Logic ($\mathcal{PL}$), and through exploiting the implicit relation between $\mathcal{PL}$ and lattice theory. In addition, we theoretically explored the merits of our model. In this chapter, we introduce different ways to build different operational instances of the previous theoretical model.

Our theoretical model presents a new way to estimate Exhaustivity and Specificity, where instead of comparing two different objects, a document d and a query q, it proposes to compare d (or q) with a part of it $d \wedge q$ (Equations 5.17 & 5.18). The theoretical model also enables us to represent different facets of d and q, where d and q can be one or several nodes in the Boolean algebra $\mathcal{B}_\Theta$ (Definitions 5.5 & 5.6).

In this chapter, we build operational instances that exploit and enable us to experimentally check the previous two advantages, namely the new approach of Exhaustivity and Specificity estimation and the possibility to map documents and queries to several nodes in $\mathcal{B}_\Theta$. We build three instances:

Exhaustivity and Specificity instance (ES).

The main goal of this instance is to check the importance of Exhaustivity and Specificity when they are explicitly integrated into a concrete IR model. We exactly consider the two new forms of Exhaustivity and Specificity (Equations 5.17 & 5.18).

This instance could be considered as the direct, or even naive, application of our model, where we do not make any special assumption about the document and query representation. We only profit from the new form of Exhaustivity and Specificity, and how they are theoretically integrated into our model, to check the importance of them in IR.

Relation-based instance (RL).

We aim in this instance to build several representations of documents and queries through exploiting some potential *semantic relations* between indexing terms. We show how it is possible to integrate an external knowledge, namely the semantic relations between terms, in our IR model.

In this instance, we mainly play on the possibility to build different representations of

documents and queries, where instead of assuming that d only corresponds to one node in $\mathcal{B}_\Theta$, we map d to several nodes through exploiting the semantic relations between terms.

Structure-based instance (ST).

The main goal of this instance is to exceed the flat representation of documents and queries through exploiting some *structural relations* between terms.

In principle, this instance is similar to the previous instance (RL). However, this instance exploits a different knowledge source, where it exceeds the flat representation of documents and queries, namely the bag of term representation, and supposes that terms are inter-related. This instance also depends on the possibility to map documents and queries to several nodes in $\mathcal{B}_\Theta$.

In each instance, we redefine documents, queries, and the matching function. However, prior to that, we must redefine the alphabet Ω through showing the nature of elements that it could contain.

This chapter is organized as follows: In section 2, we talk about the possible mapping between the alphabet or the set of atomic propositions, as a mathematical notion, and the set of indexing terms, as an IR related notion. We also talk about the type of terms that we use in this thesis and the notion of truth or falseness of a term in a particular document or query. Section 3 is dedicated to present the Exhaustivity and Specificity instance (ES) of our model, where we provide a concrete definition of documents, queries, and the matching function. In section 4, we present the relation-based instance (RL). In the same section, we also show, in a theoretical manner, that document or query expansion generally improves the recall but decreases the precision. Section 5 presents the structure-based instance (ST). We conclude in section 6.

6.2 The Alphabet

From a mathematical perspective, the alphabet Ω is a finite set of atomic propositions, whereas from an IR perspective, it is a finite set of indexing terms. Therefore, any indexing term is an atomic proposition, and in a particular document or query it can be either true or false. The truth or falseness of an indexing term in a document or a query will be discussed later in this chapter.

Normally, there are several types of indexing terms, e.g. words, concepts, phrases, etc. In this study, we mainly consider two types: *words* and *concepts*. When documents and queries are indexed using words then each atomic proposition $a_i \in \Omega$ corresponds to a word, and when they are indexed using concepts, each atomic proposition $a_i \in \Omega$ corresponds to a concept.

6.2.1 Words

Words can be defined as the smallest linguistic elements that have a semantic and can stand by themselves. Words are the classic type of indexing terms that is used to represent the content of documents and queries. For example, assume the document d is '*lobar pneumonia xray*' then the three words '*lobar*', '*pneumonia*', and '*xray*', each of them is an atomic proposition in Ω.

Most IR models suppose that words are independent, or in other words, the only relation between words is the identity, i.e. for any two words either they are identical or not. However,

transforming documents or queries to bags of words is a quite limited indexing choice, because we lose the information that comes from the order of words within the text, e.g. the meaning of the text '*The White House*' is different from the meaning of each word alone, and it also different from the meaning of the text '*the house is white*'. Therefore, besides words, we propose in this study to use concepts as another type of terms.

6.2.2 Concepts

There are many possible definitions of concepts [Bělohlávek & Klir, 2011]. In this study, concepts are defined as in (Definition 2.1–P.20). WordNet's synsets or UMLS' concepts are examples of concepts. For example, assume the document d is '*lobar pneumonia*', then in UMLS d can be mapped to two concepts '*C0032300*' and '*C0155862*'[1]. In this case, both '*C0032300*' and '*C0155862*' are atomic propositions in Ω.

In general, concepts are supposed to be more meaningful type of terms than words. There are two main advantages of using concepts instead of words. On the one hand, since concepts encompass the phrases or words that are synonymous then concepts contribute to solve the term mismatch problem, where any two synonymous words should be mapped to the same concept. On the other hand, concepts are normally a part of a knowledge resource which is supposed to contain some supplementary information about concepts, e.g. some semantic relations between concepts. Knowing that knowledge resources are normally human-validated then they are valuable resources of information. The main disadvantage of using concepts is that it is mandatory to have a tool for mapping text to concepts, and the retrieval performance of concept-based IR models is highly dependent on the precision of these tools, which are normally not very precise [Maisonnasse *et al.*, 2009].

6.2.3 Truth and Falseness of Indexing Terms

We said in (Section 4.2–P.69) that, in general, the truth or falseness of a term is related to the implicit content of documents and queries. In other words, the decision that a specific term is true or false in a specific document is not directly related to the occurrence of that term in the document.

In general, two status of truth can be identified in the relation between a document d and a term t. First, t is true in d if d contains enough information to say that d must be indexed by t. Normally, the occurrence of t in d is considered a sufficient reason for t to be true in d. Second, t is false in d if d contains enough information to say that d must not be indexed by t. Normally, the non-occurrence of t in d is considered a sufficient reason for t to be false in d. However, sometimes it is difficult to make any assumption about the truth of t in d. For example, in a document d like '*Karam lives in Grenoble*', what about '*France*'? Knowing that '*Grenoble*' is a french city, should the term '*France*' be true or false in d? By considering the Close World Assumption, any term does not occurring in d is false, whereas, by considering the Open World Assumption, we do not say anything about the truth of the term t that does not occur in d, which means, t could be true or false.

[1]The concept '*C0032300*' refers to '*lobar pneumonia*' as a disease or syndrome. Whereas the concept '*C0155862*' refers to '*streptococcal pneumonia*' as a disease or syndrome.

We distinguish between the truth or falseness of a word, in a specific document, and those of a concept, as follows:

Words. A word $a_i \in \Omega$ is considered true in a document d (or a query q) *iff* a_i occurs in d. We also consider the *open wold assumption*, where the words that do not occur in d can be considered either true or false.

Concepts. A concept $a_i \in \Omega$ is considered true in a document d (or a query q) *iff* d contains a text mapped to a_i. Here also, we consider the *open wold assumption*, where the concepts that do not correspond to any text in d can be considered either true or false.

In this manner, we do not explicitly take the falseness of a term into account. Mathematically, if $d \in \Theta_\Omega$ is a clause then the set of propositions Ω_d^- is always empty (Definition 5.3–P.80). Although, our model is capable of efficiently dealing with the cases where $\Omega_d^- \neq \emptyset$, we do not take the explicit term falseness into account, because it is very hard to automatically identify these cases in text (pure technical reason). Automatic identification of false terms is beyond the topic of this thesis.

Actually, most IR models assume that all terms occurring in d are true and all other terms are false (*close world assumption*). In this study, we consider the *open wold assumption*, where all terms occurring in d are true, but we do not make any pre-assumption about the truth or falseness of the other terms.

6.2.4 Term Weighting

The previous discussion talks about the relation between documents (or queries) and indexing terms in a binary way, where a term is either true or false in a specific document. This is a quite limited way because documents do not talk about all terms in the same way or in the same level of details. Therefore, it is mandatory to reflect this gradual nature of the relation. Moreover, all experiments in the IR field show the importance of term weighting [Fang *et al.*, 2004].

First of all, let us keep in mind that term weighting is not a part of the logic and it is a pure operational aspect, even though, it can be reflected through the degree of inclusion function (Equation 5.5–P.86). Second, assume d is a document and it is a clause $d \in \Theta_\Omega$, where $\mu(d) = (\Omega_d^+, \Omega_d^-)$. The weight of a term $a_i \in \Omega$ in a document d can be viewed as a function:

$$\forall a_i \in \Omega, \forall d \in \Theta_\Omega, \begin{cases} w_i^d > 0 & \text{if} \quad a_i \in \Omega_d^+ \\ w_i^d = 0 & \text{if} \quad a_i \in \Omega_d^- \\ w_i^d \geq 0 & \text{if} \quad a_i \in \Omega_d^\pm \end{cases} \tag{6.1}$$

The last case, $w_i^d \geq 0$ when $a_i \in \Omega_d^\pm$, will be especially seen in the relation-based instance (RL), where concepts from outside the document will take a weight because they are related to concepts from inside the document via a semantic relation.

6.3 Exhaustivity & Specificity Instance

In principle, Exhaustivity and Specificity compare either a document d with a query q or q with d. Therefore, using a symmetric uncertainty measure U will lead to an identical Exhaustivity and Specificity. Equations 5.17 & 5.18 reform the two notions, where instead of comparing

d with a totally different object q, they compare d and q with a part of them $\mu(d) \curlywedge \mu(q)$. In this manner, even using a symmetric uncertainty measure will not lead to identical Exhaustivity and Specificity, because in Exhaustivity we compare q with a part of it, whereas in Specificity we compare d with a part of it.

As we mentioned, the main goal of this instance is to check the retrieval performance of an IR model integrating the new forms of Exhaustivity and Specificity. In other words, we test the importance of Exhaustivity and Specificity.

To build an operational instance of our IR model, we need to concretely define documents, queries, and the matching function. In this instance, the alphabet Ω can be either words or concepts. In other words, this instance is applicable whatever the type of indexing terms is.

6.3.1 Documents & Queries

We follow the main assumption in most IR models, where the content of a document d is determined through the aggregation of the indexing terms that occur in it. In other words, d is a conjunction of the terms that appear in it. More precisely, d is a clause $d \in \Theta_\Omega$ (Definition 5.5 where $|\Theta_d| = 1$), and it corresponds to only one node $\mu(d)$ in the Boolean algebra $\mathcal{B}_\Theta$. The query q is represented in the same way. In this manner, the object $\mu(d) \curlywedge \mu(q)$ is also a node in $\mathcal{B}_\Theta$. More formally,

Query. Any query is a conjunction of terms and it is represented by only one clause, as follows: for any query q,

$$\exists \Omega_q \subseteq \Omega, q = \bigwedge_{a_i \in \Omega_q} a_i$$

where Ω_q is the set of terms that occur in q. Any query corresponds to only one node $\mu(q)$ in the Boolean algebra $\mathcal{B}_\Theta$, where $\mu(q) = (\Omega_q, \emptyset)$.

Document. Any document is a conjunction of terms and it is represented by only one clause, as follows:

$$\forall d \in D, \exists \Omega_d \subseteq \Omega, d = \bigwedge_{a_i \in \Omega_d} a_i$$

where Ω_d is the set of terms that occur in d. Any document corresponds to only one node $\mu(d)$ in the Boolean algebra $\mathcal{B}_\Theta$, where $\mu(d) = (\Omega_d, \emptyset)$.

The meet between them. The meet node of the two nodes $\mu(d)$ and $\mu(q)$ is:

$$\mu(d) \curlywedge \mu(q) = (\Omega_d, \emptyset) \curlywedge (\Omega_q, \emptyset) = (\Omega_d \cap \Omega_q, \emptyset)$$

For simplifying the notation, we will refer to $\mu(d) \curlywedge \mu(q)$ by $\mu(dq)$. Based on (Equations 5.17 & 5.18), instead of comparing Ω_d with Ω_q, we compare the sets Ω_d and Ω_q with the set of shared elements $\Omega_d \cap \Omega_q$, that means, we explicitly integrate the coordination level between d and q into the IR model.

Note that, for d and q, the negative terms are not considered ($\Omega_d^- = \emptyset$ and $\Omega_q^- = \emptyset$), and at the same time, we do not make any assumption about the terms that do not occur in them ($\Omega_d^\pm$ and $\Omega_q^\pm$), which means, these terms can be true or false.

6.3.2 Matching Function

For computing the matching score between a document and a query $RSV(d,q)$, the most general form (Equation 5.13–P.89) is considered. We said that each of d and q is only one clause or equivalently one node, we thus replace $G''(\{d\},\{q\})$ and $G''(\{q\},\{d\})$ by $Z(\mu(d),\mu(q))$ and $Z(\mu(q),\mu(d))$, respectively, and thus:

$$RSV_{ES}(d,q) = F\left[Z(\mu(d),\mu(q)), Z(\mu(q),\mu(d))\right]$$

According to (Equations 5.17 & 5.18), we replace each of $Z(\mu(d),\mu(q))$ and $Z(\mu(q),\mu(d))$ by $Z(\mu(dq),\mu(q))$ and $Z(\mu(dq),\mu(d))$, respectively:

$$RSV_{ES}(d,q) = F\left[Z(\mu(dq),\mu(q)), Z(\mu(dq),\mu(d))\right]$$

Now, it is mandatory to provide a concrete definition of the function Z, and to do that, $\mu(d)$, $\mu(q)$, and $\mu(dq)$ must be first redefined in a way compatible with Z.

In fact, there are many mathematical frameworks, e.g. vector space, probability, fuzzy sets, etc., to concretize our model, or in other words, to concretely define the degree of inclusion or implication function Z. Vector Space Model (VSM) is one of the earliest and simplest IR models, and also its retrieval performance is comparable to other IR models [Singhal, 2001]. Actually, we showed in (Section 5.4.3.4–P.96) the possibility to use the inner-product function to define Z. Furthermore, the inner-product satisfies the *Sum rule*, which is necessary to use the new form of Exhaustivity and Specificity. Therefore, we choose, in this thesis, to use the vector space as a mathematical framework to concretize our model. Keep in mind that the vector space is not the only choice, and there are many other choices, especially probability, because when Z is consistent with all structural properties of Boolean algebra, it becomes a conditional probability. This could be one of perspectives to develop our study.

Using (Equations 5.23 & 5.24), the three nodes $\mu(d)$, $\mu(q)$, and $\mu(dq)$ can be transformed to three vectors $\overrightarrow{\mu(d)}$, $\overrightarrow{\mu(q)}$, and $\overrightarrow{\mu(dq)}$, respectively. We know that the inner-product satisfies the *Sum rule*, then we replace the function Z by the inner-product:

$$RSV_{ES}(d,q) = F\left[\overrightarrow{\mu(dq)}\cdot\overrightarrow{\mu(q)}, \overrightarrow{\mu(dq)}\cdot\overrightarrow{\mu(d)}\right]$$

For the function F, we choose the multiplication ($\times$), because for any two values $x, y \in [0,1]$, $\lim_{x\to 0} x \times y = 0$ and $\lim_{x\to 1} x \times y = y$, and that produces a rather stable behavior because it privileges the worst case. After these choices, the matching score between a document d and a query q becomes:

$$RSV_{ES}(d,q) = \left(\overrightarrow{\mu(dq)}\cdot\overrightarrow{\mu(q)}\right)^{\alpha} \times \left(\overrightarrow{\mu(dq)}\cdot\overrightarrow{\mu(d)}\right)^{1-\alpha}$$

where $\alpha \in [0,1]$ is a tuning parameter. We introduce α to study the mutual impact between Exhaustivity and Specificity, where $\alpha = 1$ means that $RSV_{ES}(d,q)$ is totally depended on Exhaustivity, whereas $RSV_{ES}(d,q)$ is totally depended on Specificity when $\alpha = 0$. The $\log$ function is a monotonically increasing function, then it will not change the ranking if it applied to the previous equation:

$$RSV_{ES}(d,q) \propto \alpha \times \log\left(\overrightarrow{\mu(dq)}\cdot\overrightarrow{\mu(q)}\right) + (1-\alpha) \times \log\left(\overrightarrow{\mu(dq)}\cdot\overrightarrow{\mu(d)}\right)$$

According to the definition of the inner-product, the previous equation becomes:

$$RSV_{ES}(d,q) \propto \alpha \times \log\left(\sum_{a_i \in \Omega} w_i^{dq} \times w_i^q\right) + (1-\alpha) \times \log\left(\sum_{a_i \in \Omega} w_i^{dq} \times w_i^d\right) \tag{6.2}$$

This is the retrieval equation of this instance, where: w_i^q is the weight of the term a_i in the query q, w_i^d is the weight of the term a_i in the document d, and w_i^{dq} is the weight of the term a_i in the object $\mu(d) \wedge \mu(q)$.

6.4 Relation-Based Instance

Classical intersection-based IR models suffer from the term-mismatch problem, which occurs when the user who asks the query and the author of document use different terms to say the same thing. These models assume that the indexing terms are independent, which means, the only relation between terms is the identity, where any two terms are either identical or not. For example, when a user asks about the price of flats in France using the term '*flat*', then a document talking about the price of flats in France is obviously related to the query even if it uses the term '*apartment*', because the two terms '*flat*' and '*apartment*' are synonymous.

Intersection-based IR models also suffer from another problem, which appears when the query asks about something very general and the document contains information about another thing that can be considered as a specification of the query's thing. For example, assume a user searching an article about '*trees*', then in this case, an article about '*cypress*' or about '*pine*' is probably related to the query, because $cypress \xrightarrow{\text{isa}} tree$ and $pine \xrightarrow{\text{isa}} tree$.

Another problem also appears when a query is about something and a document is about a part of that thing, and vice-versa. For example, if a user asking for some information about '*fingers*', then a document containing information about the '*hand*' is susceptible to have the information that the user is looking for, because $finger \xrightarrow{\text{part-of}} hand$.

The previous discussion shows that terms are normally not independent, and it also shows the importance of exploiting the semantic relations between terms. Semantic relations can link concepts or even words. In this instance, we choose concepts to build our index, because concepts are normally a part of a knowledge resource, which is supposed to contain rich information about concepts and the semantic relations between them. Anyway, this instance is also applicable to words, but we prefer to currently restrict it to concepts. One more reason to focus on concepts rather than words is that concepts normally encompass all synonymous words and phrases. Thus, concepts automatically contribute to solve the term-mismatch problem.

6.4.1 Documents & Queries

We assume that in this instance the elements of Ω are restricted to concepts. Concepts are a part of a knowledge resource, which contains supplementary information about concepts and defines semantic relations between them. Relations between concepts are normally directed and labeled, where $a_i \xrightarrow{r} a_j$ is different from $a_i \xrightarrow{r'} a_j$ and $a_j \xrightarrow{r} a_i$. Let us refer to the concept a_j that is related to the concept a_i via a relation r directed from a_i to a_j by a_i^r. In other words, $a_i^r = a_j$ where $a_i \xrightarrow{r} a_j$.

Assume that the original document d^o is a clause $d^o \in \Theta_\Omega$. There are several choices to extend d^o using concepts related to the original concepts of d^o via semantic relations. We present some of these choices through an example. Assume $d^o = a_1 \wedge a_2$, and assume a_1^r and a_2^r are the two concepts related to a_1 and a_2 via the relation r, respectively. The new document d after expanding d^o by the relation r becomes:

Choice 1. Considering either a_i or its related concept a_i^r. In this case, the document d becomes: $d = (a_1 \vee a_1^r) \wedge (a_2 \vee a_2^r)$ or equivalently $d = (a_1 \wedge a_2) \vee (a_1^r \wedge a_2) \vee (a_1 \wedge a_2^r) \vee (a_1^r \wedge a_2^r)$. On the one hand, this choice transforms the original document d^o from only one node in $\mathcal{B}_\Theta$ to a huge number of nodes. On the other hand, this choice is not very reasonable because choosing among something belongs to the original document d^o and another thing from outside of d^o, means that, the outer thing can replace the original one, and that could lead to a lot of noise. Note that the last clause in the previous equation $(a_1^r \wedge a_2^r)$ does not contain anything from the original document d^o.

Choice 2. Building two versions, representing two points of view, of the original document d^o, one with only the original concepts (a_1 and a_2) and another with only the related concepts (a_1^r and a_2^r). In this case, the document d becomes: $d = d^o \vee d^r$ where $d^r = a_1^r \wedge a_2^r$. This choice will lead to noisy representation of d, where the second clause in the previous equation $(a_1^r \wedge a_2^r)$ is so far from the original document d^o.

Choice 3. Instead of choosing among a concept and its related concept, we consider both at the same time, that means, extending the original document d^o by new concepts using a specific semantic relation. The document d becomes: $d = d^o \vee d^r$ where in this case $d^r = a_1 \wedge a_1^r \wedge a_2 \wedge a_2^r$.

On the one hand, when $d \supset q$ is valid then $d^o \supset q$ and $d^r \supset q$ are also valid, because $\models d \supset q$ can be rewritten as $\models [(d^o \supset q) \wedge (d^r \supset q)]$. On the other hand, assume that the relation $a_i \xrightarrow{r} a_j$ is represented by $\models a_i \supset a_j$ [Chiaramella & Chevallet, 1992], then d^r is resulted from d^o through revising it using the knowledge $\Gamma = \{\models a_i \supset a_j, \dots\}$, where $d^o \wedge (a_i \supset a_j) = (d^o \wedge \neg a_i) \vee (d^o \wedge a_j) = d^o \wedge a_j$ because $a_i \in d^o$ and thus $d^o \wedge \neg a_i$ is unsatisfiable or always false;

Note that in the above choices, we only talk about documents without referring to queries, although queries are an essential part of the retrieval process. The intrinsic goal of IR is to answer a query, so what is the benefit of extending a document by some concepts non-related to the query? Assume that the query q is about a_1^r but not about a_2^r, then adding a_2^r to the original document d^o will only increase the noise. By considering the third choice and by taking the query q into account, the document d becomes: $d = d^o \vee d^r$ where $d^r = a_1 \wedge a_1^r \wedge a_2$.

Using semantic relations to expand a document lead to better recall, but at the same time it could decrease the precision. For example, assume that the query asks for '*rock*' as a type of music, then extending the document that talks about '*quartz*' by the concept '*rock*' will lead to noise. Remember that $quartz \xrightarrow{\text{isa}} rock$. Furthermore, using relations could lead to add some non-informative or very general concepts, where everything *is an* entity. For example, in UMLS: $Bcell \xrightarrow{\text{isa}} Cell \xrightarrow{\text{isa}} FullyFormedAnatomicalStructure \xrightarrow{\text{isa}} AnatomicalStructure \xrightarrow{\text{isa}} PhysicalObject \xrightarrow{\text{isa}} Entity$. Actually, here lies the role of term weighting, which should maintain the recall improvement and avoid the decrease in precision. More formally, documents and queries are:

Query. Any query is a conjunction of concepts and it is represented by only one clause, as follows: for any query q,

$$\exists \Omega_q \subseteq \Omega, q = \bigwedge_{a_i \in \Omega_q} a_i$$

where Ω_q is the set of concepts that the text of q is mapped to. Any query corresponds to only one node $\mu(q)$ in the Boolean algebra $\mathcal{B}_\Theta$, where $\mu(q) = (\Omega_q, \emptyset)$.

Document. Any document d can be written as a disjunction of two documents: d^o the original document and d^r the expanded version of d^o using the semantic relation r.

$$d = d^o \vee d^r$$

The original document d^o is a conjunction of concepts and it is represented by only one clause, as follows:

$$\exists \Omega_d^o \subseteq \Omega, d^o = \bigwedge_{a_i \in \Omega_d^o} a_i$$

where Ω_d^o is the set of concepts that the text of the original document d^o is mapped to. The expanded document d^r is also a conjunction of concepts and it is represented by only one clause, as follows:

$$\exists \Omega_d^r \subseteq \Omega, d^r = \bigwedge_{a_i \in \Omega_d^r} a_i$$

where $\Omega_d^r = \Omega_d^o \cup \Omega_r$ and Ω_r contains the expansion terms that are connected to the terms of Ω_d^o via the relation r:

$$\Omega_r = \{a_j \in \Omega \setminus \Omega_d^o | \exists a_i \in \Omega_d^o, a_i \xrightarrow{r} a_j\}$$

The original document d^o corresponds to only one node $\mu(d^o)$ in the Boolean algebra $\mathcal{B}_\Theta$, where: $\mu(d^o) = (\Omega_d^o, \emptyset)$. The expanded document d^r also corresponds to one node $\mu(d^r)$ in $\mathcal{B}_\Theta$, where $\mu(d^r) = (\Omega_d^r, \emptyset)$. Note that whatever is the relation r, $\mu(d^o) \leq \mu(d^r)$. Therefore, the document d corresponds to two comparable nodes $\mu(d^o)$ and $\mu(d^r)$ in the Boolean algebra $\mathcal{B}_\Theta$.

It is possible to build several extensions of the original document d^o using different semantic relations r_i. In this case, the final document d becomes: $d = d^o \vee d^{r_1} \vee \cdots \vee d^{r_k}$. Note that d corresponds to several nodes in $\mathcal{B}_\Theta$, one node for each relation and the node of the original document.

The meet between them. The meet node of the node of the original document $\mu(d^o)$ and the node of the query $\mu(q)$ is:

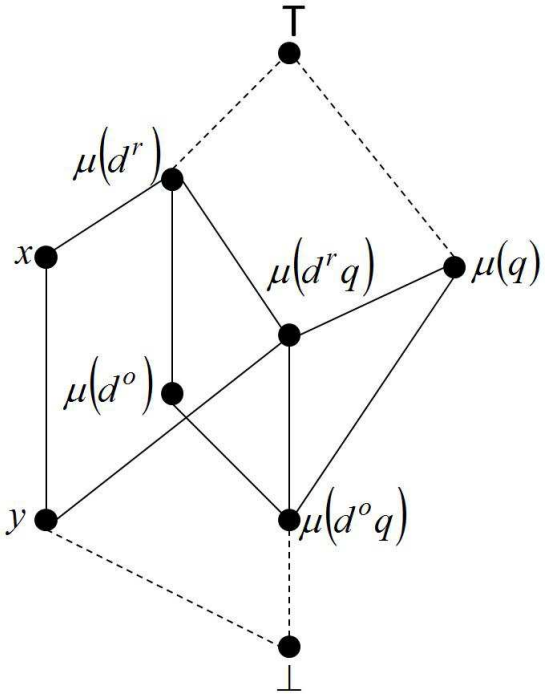
$$\mu(d^o q) = \mu(d^o) \curlywedge \mu(q) = (\Omega_d^o, \emptyset) \curlywedge (\Omega_q, \emptyset) = (\Omega_d^o \cap \Omega_q, \emptyset)$$

The meet node of the node of the expanded document $\mu(d^r)$ and the node of the query $\mu(q)$ is:

$$\mu(d^r q) = \mu(d^r) \curlywedge \mu(q) = (\Omega_d^r, \emptyset) \curlywedge (\Omega_q, \emptyset) = (\Omega_d^r \cap \Omega_q, \emptyset)$$

Note also that the coordination level between d and q becomes bigger, where $\mu(d^o q) \leq \mu(d^r q)$, and that is exactly what contribute to solve the term mismatch problem.

Figure 6.1: Document expansion using a semantic relation r w.r.t. $\mathcal{B}_\Theta$.



The position of $\mu(d^o)$, $\mu(d^r)$, $\mu(q)$, $\mu(d^o q)$, and $\mu(d^r q)$ on the Boolean algebra $\mathcal{B}_\Theta$ is depicted in (Figure 6.1). Note that, for d^o and q, the negative terms are not considered ($\Omega^-_{d^o} = \emptyset$ and $\Omega^-_q = \emptyset$), and at the same time, we do not make any assumption about the terms that do not appear in them ($\Omega^\pm_{d^o}$ and $\Omega^\pm_q$), which means, these terms can be true or false.

6.4.2 Matching Function

Before talking about the matching function, let us *theoretically* study the difference between $RSV(d^o, q)$ and $RSV(d^r, q)$. We discuss the difference from two points of view:

Exhaustivity. In order to show the difference between $RSV(d^o, q)$ and $RSV(d^r, q)$ from Exhaustivity point of view, it is sufficient to study the difference between $Z(\mu(d^o), \mu(q))$ and $Z(\mu(d^r), \mu(q))$.

We know that $\mu(d^o) \leq \mu(d^r)$ and $\mu(d^o q) \leq \mu(d^r q) \leq \mu(q)$ (Figure 6.1). Therefore, we have:

$$\begin{aligned} Z(\mu(d^o), \mu(q)) &= Z(\mu(d^o q), \mu(q)) && \text{(Equation 5.17)} \\ &= Z(\mu(d^o q), \mu(d^r q)) \times Z(\mu(d^r q), \mu(q)) && \text{(Equation B.9)} \\ &= Z(\mu(d^o q), \mu(d^r q)) \times Z(\mu(d^r), \mu(q)) && \text{(Equation 5.17)} \end{aligned}$$

From the definition of the Z function, we know that: $Z(\mu(d^o q), \mu(d^r q)) \leq 1$, and hence:

$$Z(\mu(d^o), \mu(q)) \leq Z(\mu(d^r), \mu(q)) \tag{6.3}$$

Specificity. In order to show the difference between $RSV(d^o, q)$ and $RSV(d^r, q)$ from Specificity point of view, it is sufficient to study the difference between $Z(\mu(q), \mu(d^o))$ and $Z(\mu(q), \mu(d^r))$.

We know that $\mu(d^o q) \leq \mu(d^o) \leq \mu(d^r)$ (Figure 6.1). Therefore, we have:

$$\begin{aligned} Z(\mu(d^o q), \mu(d^r)) &= Z(\mu(d^o q), \mu(d^o)) \times Z(\mu(d^o), \mu(d^r)) \quad \text{(Equation B.9)} \\ &= Z(\mu(q), \mu(d^o)) \times Z(\mu(d^o), \mu(d^r)) \quad \text{(Equation 5.18)} \end{aligned}$$

We also know that $\mu(d^o q) \leq \mu(d^r q) \leq \mu(d^r)$ (Figure 6.1). Therefore, we have:

$$\begin{aligned} Z(\mu(d^o q), \mu(d^r)) &= Z(\mu(d^o q), \mu(d^r q)) \times Z(\mu(d^r q), \mu(d^r)) \quad \text{(Equation B.9)} \\ &= Z(\mu(d^o q), \mu(d^r q)) \times Z(\mu(q), \mu(d^r)) \quad \text{(Equation 5.18)} \end{aligned}$$

Finally, by comparing the previous two values of $Z(\mu(d^o q), \mu(d^r))$, we get:

$$\frac{Z(\mu(q), \mu(d^o))}{Z(\mu(q), \mu(d^r))} = \frac{Z(\mu(d^o q), \mu(d^r q))}{Z(\mu(d^o), \mu(d^r))} \tag{6.4}$$

In general, we cannot conclude if $Z(\mu(q), \mu(d^o)) \leq Z(\mu(q), \mu(d^r))$ or vice-versa. However, it is possible to choose two nodes $x = (\Omega_r, \emptyset)$ and $y = (\Omega_r \cap \Omega_q, \emptyset)$ from $\mathcal{B}_\Theta$. The two nodes x, y satisfy: $\mu(d^r) = \mu(d^o) \veebar x$ and $\mu(d^r q) = \mu(d^o q) \veebar y$. At the same time, we have: $y \leq x$ (Figure 6.1). That means, the difference between $\mu(d^o)$ and $\mu(d^r)$ is bigger than the difference between $\mu(d^o q)$ and $\mu(d^r q)$. Hence, it is more probable that $Z(\mu(q), \mu(d^o)) \geq Z(\mu(q), \mu(d^r))$.

It is known that Exhaustivity is recall-oriented whereas Specificity is precision-oriented [Losada & Barreiro, 2001]. Therefore, from (Equations 6.3 & 6.4) we can conclude that document expansion improves the recall, but at the same time, we can not draw a clear conclusion about the precision, even though the precision normally decreases which is clear from the fact that $y \leq x$. This discussion provides a mathematical formalization of the well accepted idea in IR that claims: document or query expansion increases recall but decreases precision. Actually, it is a game of term weighting to maintain recall increasing and avoid precision decreasing.

After clarifying the difference between $RSV(d^o, q)$ and $RSV(d^r, q)$, now it is the time to show how $RSV_{RL}(d, q)$ can be computed. The document d corresponds to two nodes $\mu(d^o)$ and $\mu(d^r)$ in the Boolean algebra $\mathcal{B}_\Theta$. The query q corresponds to one node $\mu(q)$ in $\mathcal{B}_\Theta$. According to the most general form of matching function (Equation 5.13), we have:

$$\begin{aligned} RSV_{RL}(d, q) &= F\left[G''\left(\{d^o, d^r\}, \{q\}\right), G''\left(\{q\}, \{d^o, d^r\}\right)\right] \\ &= F\left[G'\left(Z(\mu(d^o), \mu(q)), Z(\mu(d^r), \mu(q))\right), G\left(Z(\mu(q), \mu(d^o)), Z(\mu(q), \mu(d^r))\right)\right] \end{aligned}$$

We choose multiplication $\times$ to replace G', because G' must be a triangular norm function. Whereas, we choose addition $+$ to replace G, because G must be a triangular conorm function. Concerning the function F, we choose multiplication $\times$. In view of these choices, the matching function becomes:

$$\begin{aligned} RSV_{RL}(d, q) &= Z(\mu(d^o), \mu(q)) \times Z(\mu(d^r), \mu(q)) \times [Z(\mu(q), \mu(d^o)) + Z(\mu(q), \mu(d^r))] \\ &= Z(\mu(d^o), \mu(q)) \times Z(\mu(d^r), \mu(q)) \times Z(\mu(q), \mu(d^o)) \\ &\quad + Z(\mu(d^o), \mu(q)) \times Z(\mu(d^r), \mu(q)) \times Z(\mu(q), \mu(d^r)) \\ &= Z(\mu(d^o q), \mu(q)) \times Z(\mu(d^r q), \mu(q)) \times Z(\mu(d^o q), \mu(d^o)) \\ &\quad + Z(\mu(d^o q), \mu(q)) \times Z(\mu(d^r q), \mu(q)) \times Z(\mu(d^r q), \mu(d^r)) \end{aligned}$$

If we choose to replace Z by inner-product and the other components by vectors as in the previous model $RSV_{ES}(d,q)$ then:

$$\begin{aligned} RSV_{RL}(d,q) &= \left(\sum_{a_i\in\Omega} w_i^{d^o q} \times w_i^q\right) \times \left(\sum_{a_i\in\Omega} w_i^{d^r q} \times w_i^q\right) \times \left(\sum_{a_i\in\Omega} w_i^{d^o q} \times w_i^{d^o}\right) \\ &+ \left(\sum_{a_i\in\Omega} w_i^{d^o q} \times w_i^q\right) \times \left(\sum_{a_i\in\Omega} w_i^{d^r q} \times w_i^q\right) \times \left(\sum_{a_i\in\Omega} w_i^{d^r q} \times w_i^{d^r}\right) \end{aligned}$$

where, w_i^q is the weight of the concept a_i in the query q, $w_i^{d^o}$ is the weight of the concept a_i in the original document d^o, $w_i^{d^r}$ is the weight of the concept a_i in the expanded document d^r, $w_i^{d^o q}$ is the weight of the concept a_i in the object $\mu(d^o) \curlywedge \mu(q)$, and $w_i^{d^r q}$ is the weight of the concept a_i in the object $\mu(d^r) \curlywedge \mu(q)$.

We know that, $\mu(d^r q) = (\Omega_d^r \cap \Omega_q, \emptyset)$, we can thus estimate $w_i^{d^r q}$ depending on the concepts of q. In addition, we illustrated that document expansion, using a semantic relation r, increases the recall but it can decrease the precision if it is not carefully considered. To avoid the noise that comes from the document expansion, we weight the concepts of d^r through the weight of the corresponding concepts in d^o. First, we need a measure to estimate the semantic similarity between any two concepts having a semantic relation r, as follows:

$$\forall a_i, a_j \in \Omega, Sim_r(a_i,a_j) = \begin{cases} 1 & \text{if} \quad a_i = a_j \\ 0 & \text{no semantic relation} \\ 0 < u < 1 & \text{if} \quad a_i \xrightarrow{r} a_j \end{cases} \tag{6.5}$$

The Sim_r measure reflects the fact that the strongest relation between concepts is the identity, and the weakest is that they are totally different, whereas if two concepts are linked by any type of semantic relations then the measure should reflect the nature of this relation and the way the two concepts are linked via this relation. Second, the weight of concepts in the expanded document d^r becomes:

$$\forall a_i \in \Omega, w_i^{d^r} = Sim_r(a_j, a_i) \times w_j^{d^o}$$

where, a_j is the concept of the original document d^o that maximizes the similarity measure of the semantic relation r. In other words,

$$a_j = \arg\max_{a_k \in \Omega_d^o} Sim_r(a_k, a_i)$$

The final equation to compute the matching score between q and d becomes:

$$\begin{aligned} RSV_{RL}(d,q) &= \left(\sum_{a_i\in\Omega} w_i^{d^o q} \times w_i^q\right) \times \left(\sum_{a_i\in\Omega} w_i^{d^r q} \times w_i^q\right) \times \left(\sum_{a_i\in\Omega} w_i^{d^o q} \times w_i^{d^o}\right) \\ &+ \left(\sum_{a_i\in\Omega} w_i^{d^o q} \times w_i^q\right) \times \left(\sum_{a_i\in\Omega} w_i^{d^r q} \times w_i^q\right) \times \left(\sum_{a_i\in\Omega} w_i^{d^r q} \times Sim_r(a_j, a_i) \times w_j^{d^o}\right) \end{aligned} \tag{6.6}$$

6.5 Structure-Based Instance

Flat document and query representation is a quite limited way to represent the content of documents and queries, because it is not capable of expressing the potential relations between

Table 6.1: The phrase '*lobar pneumonia xray*' and its corresponding concepts in UMLS.

The phrase and its parts	Corresponding concepts
'*lobar pneumonia xray*'	-
'*lobar pneumonia*'	*C0032300, C0155862*
'*pneumonia xray*'	*C0581647*
'*pneumonia*'	*C0024109, C1278908, C0032285, C2707265, C2709248*
'*lobar*'	*C1522010, C1428707, C0796494*
'*xray*'	*C0034571, C0043299, C0043309, C1306645, C1714805 C1962945*

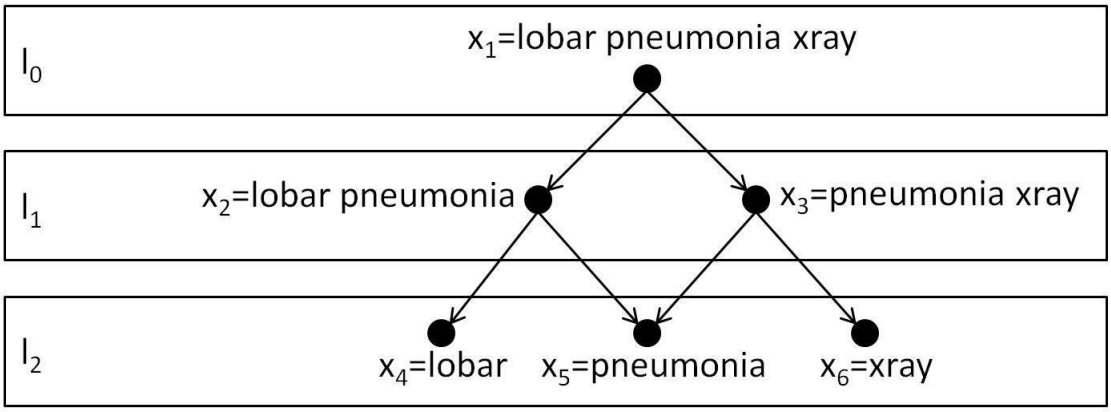
terms. For example, representing documents and queries as a bag of terms will not differentiate between '*The White House*' and '*the house is white*'. The order of terms and the structure of phrases are important components of the content of documents and queries. Conceptual indexing is supposed to be capable of overcoming this problem, because the first step in the conceptual indexing process is to identify noun-phrases in the text, and then try to map them to some concepts in a knowledge resource.

However, for a noun-phrase, it is sometimes difficult to find concepts corresponding to the whole noun-phrase. Moreover, even if there are concepts corresponding to the whole noun-phrase, it is useful to return some concepts corresponding to parts of it, because restricting our attention to the concepts that only correspond to the whole phrase could lead to miss some related concepts, or in other words, it could lead to loss in recall. For example, (Table 6.1) shows the UMLS' concepts that correspond to the phrase '*lobar pneumonia xray*' or to a part of it. You can see that the original phrase does not correspond to any concept and that justifies searching concepts that correspond to a part of it. You can also see that by replacing the phrase '*pneumonia xray*' by the concept '*C0581647*' without considering the concepts that correspond to '*pneumonia*' or to '*xray*' separately, we get a recall problem, because a document about '*pneumonia xray*' will not be able to answer a query about '*xray*' or '*pneumonia*'.

From the above example, we can see that a disambiguation step to choose one concept among the candidate concepts to represent the phrase is inadequate and can decrease the recall [Sanderson, 1994]. Moreover, transforming the phrase to a simple and flat set of the candidate concepts will decrease the precision. Fortunately, it is not difficult to build a hierarchical structure of concepts depending on the parts of phrase that theses concepts correspond to (Figure 6.2) [Bruza & van der Gaag, 1993].

Furthermore, another type of problems arises when mapping each part, e.g. x_2, to concepts. Normally, each phrase or sub-phrase is mapped to several candidate concepts, but only one of these concepts is supposed to be the most appropriate to the context of the phrase, and this is actually the role of the disambiguation step of the conceptual indexing process to select the most appropriate concept among the candidate concepts. Therefore, we think that the appropriate logical sentence to represent x_2 is $x_2 = C0032300 \vee C0155862$, and not, $x_2 = C0032300 \wedge C0155862$ (Table 6.1). This behavior, on the one hand, explosively increases the number of clauses that are necessary to represent the document d. On the other hand, this behavior is totally different from the behavior of words, where the potential relation between the words of a document is the conjunction. There are two possible approaches to deal with this behavior.

Figure 6.2: The hierarchical structure of the phrase '*lobar pneumonia xray*'.



First, it is possible to pass again through a disambiguation step, but disambiguation is normally so difficult process. In addition, even if it is so precise and accurate, studies show that a precise disambiguation will negligibly improve the retrieval performance [Sanderson, 1994]. Second, we can convert the disjunction between concepts to conjunction, but with proposing a new concept weighting system that implicitly reflects the disjunction between concepts [Abdulahhad *et al.*, 2012b, 2013b].

Finally, we must clearly say that this instance is only applicable to concepts, or in other words, Ω is a set of concepts.

6.5.1 Documents & Queries

Let us assume that any document d or query q is a list of phrases, and each phrase has a hierarchical structure of its parts or sub-phrases. Assume that each part is a logical sentence. Considering (Figure 6.2), there are several choices to represent documents and queries, among them:

Choice 1. Assume the document d is a list of two phrases s_1 and s_2 then: $d = s_1 \wedge s_2$. If each phrase has a hierarchical structure as in (Figure 6.2) then $s_1 = (x_1) \vee (x_2 \wedge x_3) \vee (x_4 \wedge x_5 \wedge x_6)$, because each level is supposed to be sufficient to represent the phrase. The problem in this way of representation is that d will correspond to huge number of clauses or nodes in the Boolean algebra $\mathcal{B}_\Theta$.

Choice 2. Instead of seeing the document d as a list of phrases, it is possible to see it as different levels of abstraction $d = l_0 \vee l_1 \vee l_2$. Each level can be constructed as a clause, e.g. $l_1 = x_2 \wedge x_3$. In this manner, we reduce the number of nodes of $\mathcal{B}_\Theta$ that are necessary to represent the document. The problem in this choice is that the high levels will suffer from low-recall problem and the low levels will contain a lot of noise.

Choice 3. We see the document as in choice 2, but each level will be a conjunction of itself and the levels below it, e.g. $l_1 = x_2 \wedge x_3 \wedge x_4 \wedge x_5 \wedge x_6$. It is clear that in this case the highest level will contain all other levels.

In this instance, we consider the third choice, where $d = l_0 = x_1 \wedge x_2 \wedge x_3 \wedge x_4 \wedge x_5 \wedge x_6$. However, we will reflect the structure of phrases and the disjunctive relation between concepts

through a new concept weighting system, which we present in (Appendix A–P.165). More formally,

Query. Any query is a conjunction of concepts and it is represented by only one clause, as follows: for any query q,

$$\exists \Omega_q \subseteq \Omega, q = \bigwedge_{a_i \in \Omega_q} a_i$$

where Ω_q is the set of concepts that the text of q is mapped to. Any query corresponds to only one node $\mu(q)$ in the Boolean algebra $\mathcal{B}_\Theta$, where: $\mu(q) = (\Omega_q, \emptyset)$.

Document. Any document is a conjunction of concepts and it is represented by only one clause, as follows:

$$\forall d \in D, \exists \Omega_d \subseteq \Omega, d = \bigwedge_{a_i \in \Omega_d} a_i$$

where Ω_d is the set of concepts that the text of d is mapped to. Any document corresponds to only one node $\mu(d)$ in the Boolean algebra $\mathcal{B}_\Theta$, where: $\mu(d) = (\Omega_d, \emptyset)$.

The meet between them. The meet node of the node of the document $\mu(d)$ and the node of the query $\mu(q)$ is:

$$\mu(dq) = \mu(d) \curlywedge \mu(q) = (\Omega_d, \emptyset) \curlywedge (\Omega_q, \emptyset) = (\Omega_d \cap \Omega_q, \emptyset)$$

Note that, for d and q, the negative terms are not considered ($\Omega_d^- = \emptyset$ and $\Omega_q^- = \emptyset$), and at the same time, we do not make any assumption about the terms that do not appear in them ($\Omega_d^\pm$ and $\Omega_q^\pm$), which means, these terms can be true or false.

6.5.2 Matching Function

The definition of documents and queries in this instance is the same as in the first instance $RSV_{ES}(d, q)$, we thus use the same matching equation:

$$RSV_{ST}(d, q) \propto \alpha \times \log \left(\sum_{a_i \in \Omega} w_i^{dq} \times w_i^q \right) + (1 - \alpha) \times \log \left(\sum_{a_i \in \Omega} w_i^{dq} \times w_i^d \right) \tag{6.7}$$

The main difference between this instance and the first one is that here we use concepts instead of words and we suppose that there is a hierarchical structure between concepts. The central point in this instance is the new concept weighting system that we claim its ability to reflect the hierarchical structure.

6.6 Conclusion

This chapter can be considered as a link between the theoretical model, which is presented in (Chapter 5–P.77), and the concrete and operational models that can be implemented and tested. Before talking about the operational IR instances proposed in this chapter, we had presented how the alphabet Ω can be transformed from the mathematical world, where Ω is a set of atomic

propositions, to the IR world, where Ω can be seen as a set of indexing terms. In this study, we propose to use two types of indexing terms: words and concepts.

We presented, in this chapter, three operational IR instances inspired from the theoretical IR model presented in the previous chapter. The first instance $RSV_{ES}(d, q)$ (Equation 6.2) considers the flat document and query representation. This instance is applicable whatever the type of terms is.

The second instance $RSV_{RL}(d, q)$ (Equation 6.6) claims that terms are not independent. The instance exploits the semantic relations between terms to expand documents, in order to overcome the term-mismatch problem. This instance is applicable to words or concepts, but, in this study, we restrict the instance to only deal with concepts. In the context of this instance, we theoretically show that document or query expansion generally improves the recall but decrease the precision.

The third instance $RSV_{ST}(d, q)$ (Equation 6.7) deals with the inadequacy of flat document and query representation, where it claims that there is a type of hierarchical relation between terms. Actually, this instance does not explicitly express this hierarchy, instead of that, it uses a new weighting system, which is presented in (Appendix A–P.165), being capable of reflecting or modeling this hierarchical structure. This instance with its current configuration is only applicable to concepts.

We investigated three instances of our theoretical model. In all instances the proposed sentences to represent documents and queries are fairly simple, where in most cases these sentences are restricted to clauses without even taking the negative form of propositions into account (for any s, $\Omega_s^- = \emptyset$). Our model is capable of efficiently dealing with any logical sentence. However, we choose to represent documents and queries in this simple way because the automatic construction of more complex and expressive logical sentences based on the textual content is not an easy and evident process, especially identifying the negative form of propositions. The automatic transformation of the content of documents and queries to expressive logical sentences, even in a simple logic like $\mathcal{PL}$, is still an open research question and it is beyond the topic of this thesis [Kim *et al.*, 2011; Losada & Barreiro, 2003].

Part IV

EXPERIMENTS

Chapter 7

Experimental Setup

7.1 Introduction

Any new Information Retrieval (IR) model needs to be tested and compared with other state of the art models, to finally establish valuable and meaningful conclusions about its effectiveness. However, the goal of experimentally testing the instances of our model is twofold. First, showing that our proposed model is applicable to large-scale data, and it can be efficiently tested on corpora of big sizes. Actually, we aim to show that our model satisfies the main thesis' goal that is announced in (Chapter 1–P.3). Second, comparing the retrieval performance of the instances of our model with some high-performance IR models, in order to place our model with respect to the state-of-the-art models, and to draw some valuable conclusions about how to improve the instantiating process of our model.

In (Chapter 5–P.77), we presented our theoretical IR model. In (Chapter 6–P.101), we built three operational instances of our theoretical model. In this chapter, we present the experimental framework and all technical details that allow us to test the three instances, and to compare their retrieval performance with the performance of some high-performance models.

Nowadays, users of IR systems search information of different modalities, e.g. text, images, videos, etc. In this thesis, we only deal with text. Therefore, we apply our model to documents and queries of textual content.

The chapter is organized as follows: Section 2 presents the way in which we extract words from the text, and also how we map the textual content to concepts. In the same section, we also talk about the semantic relations that we use in this thesis and how we compute the semantic similarity between two concepts. In section 3, we present our ways of weighting, where besides the classical weighting schema (a variant of *tf.idf*), we use a new way of concept counting, namely the Relative Concept Frequency (RCF). Section 4 is dedicated to describe the corpora, which we apply our model to, and their basic statistics. Section 5 reviews the metrics that we use for comparing the retrieval performance of IR models. In section 6 we present the baseline models that are used for comparison purposes. We conclude in Section 7.

7.2 Indexing Terms Definition

As we said in (Chapter 6–P.101), we use two types of indexing terms: *words* and *concepts*. Of course we use words, as a usually-used type of terms, to represent the content of documents and queries. However, although the proved effectiveness of using words as indexing terms, we propose to use concepts, which are supposed to be more informative than words. In fact, since each concept is the identifier of a category that encompasses all synonymous phrases (Definition 2.1–P.20), concepts implicitly contribute to solve the term-mismatch problem. Furthermore, concepts are a part of a knowledge resource, which contains supplementary information that can be exploited besides concepts, e.g. semantic relations.

7.2.1 Words

Words are classically extracted from text after removing stop words and stemming. In this study, we eliminate stop words and stem the remaining words using *Porter* algorithm [Porter, 1997] to finally get the list of words that index documents and queries. We use the SMART stop words list [Salton, 1971].

When using words as indexing terms, the alphabet or the set of atomic propositions Ω becomes a set of words, where any word can be true or false in a particular document or query (Section 6.2–P.102).

7.2.2 Concepts

Besides words, we propose to use a more informative type of terms, namely concepts. Normally, concepts are a part of a knowledge resource, and they are connected via a variety of semantic relations. Concepts do not originally belong to documents and queries like words, then we need a tool to map the textual content of documents and queries to concepts.

The concepts of many knowledge resources can be used in IR, e.g. the synsets of WordNet, the concepts of UMLS, etc. In this study, we choose UMLS as a source of concepts in the medical field. Therefore, we need a tool to map text to UMLS concepts.

For moving from text to concepts, there are many tools [Aronson, 2006; Baziz, 2005; Dozier *et al.*, 2007; Maisonnasse, 2008; Maisonnasse *et al.*, 2009], which essentially map text to concepts of a knowledge resource. Each tool proposes some concepts for a certain piece of text, and some of these tools also achieve a supplementary step to filter or disambiguate the proposed concepts for providing a more precise list of concepts [Baziz, 2005]. In this study, and since we use UMLS as a concepts source, we use MetaMap [Aronson, 2006] that provides the basic mapping functionality, and maps medical text to UMLS concepts.

In fact, the textual content of documents and queries is mapped to UMLS concepts using MetaMap. In this case, the alphabet or the set of atomic propositions Ω becomes a set of concepts, where any concept can be true or false in a particular document or query (Section 6.2–P.102). Figure 7.1 shows the output of MetaMap when it is applied to the phrase '*lobar pneumonia*', where we can see the candidate concepts of the whole phrase and its sub-phrases. We should mention here that we maintain all candidate concepts and not only the mapping ones.

Figure 7.1: MetaMap's output of the text '*lobar pneumonia*'

Phrase: "lobar pneumonia"

Meta Candidates (11):
1000 C0032300:Lobar Pneumonia [Disease or Syndrome]
1000 C0155862:Lobar pneumonia (Streptococcal pneumonia) [Disease or Syndrome]
917 C0225752:lobe lung (Structure of lobe of lung) [Body Part, Organ, or Organ Component]
861 C0032285:Pneumonia [Disease or Syndrome]
861 C1522010:Lobar [Qualitative Concept]
789 C0796494:Lobus (lobe) [Body Part, Organ, or Organ Component]
789 C1428707:Lobe (AKT1S1 gene) [Gene or Genome]
768 C0024109:Lung [Body Part, Organ, or Organ Component]
768 C1278908:Lung (Entire lung) [Body Part, Organ, or Organ Component]
755 C2707265:Pulmonary (Pulmonary:-:Point in time:^Patient:-) [Clinical Attribute]
755 C2709248:Pulmonary (Pulmonary (qualifier value)) [Qualitative Concept]

Meta Mapping (1000):
1000 C0032300:Lobar Pneumonia [Disease or Syndrome]

Meta Mapping (1000):
1000 C0155862:Lobar pneumonia (Streptococcal pneumonia) [Disease or Syndrome]

7.2.3 Semantic Relations

When using concepts, we implicitly take *synonymy* into account, because each concept encompasses all synonymous phrases (Definition 2.1–P.20). However, concepts are also interconnected via semantic relations, e.g. antonymy (opposition), hyponymy-hypernymy (specific-general), meronymy-holonymy (part-whole), etc.

The relation-based (RL) instance of our model exploits semantic relations between concepts. Therefore, there are two issues we need to decide about: which relation will we concretely use? and how do we estimate the semantic similarity between two concepts connected via this relation? namely the concrete definition of (Equation 6.5–P.112).

The semantic relation that will be used in the RL instance of our model is the 'isa' relation (Hyponymy / Hypernymy) between concepts. This relation is defined on the concepts of UMLS. The isa relation is:

- Transitive: for any three concepts c_1, c_2, and c_3 if $c_1 \xrightarrow{isa} c_2$ and $c_2 \xrightarrow{isa} c_3$ then $c_1 \xrightarrow{isa} c_3$.
- Anti-symmetric: if $c_1 \xrightarrow{isa} c_2$ and $c_2 \xrightarrow{isa} c_1$ then $c_1 = c_2$.

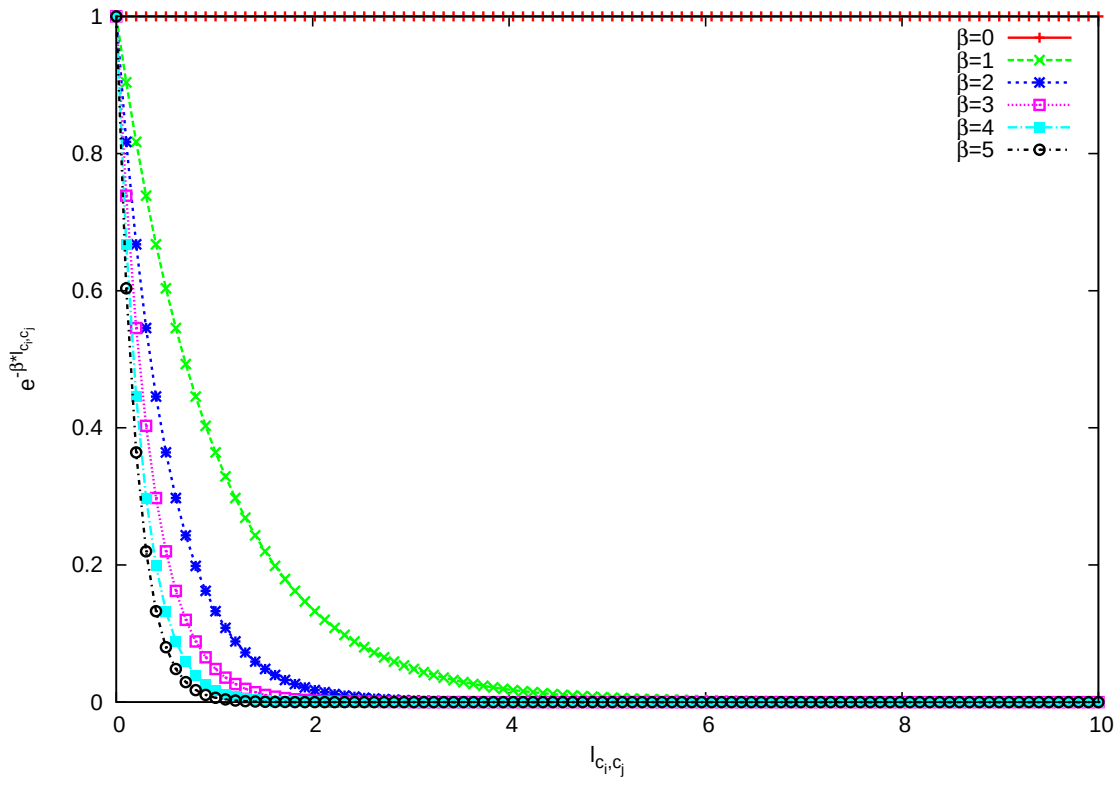
Therefore, it is possible to define the length or the number of edges $l_{c_i.c_j}$ of an isa-path between two concepts c_i and c_j directed from c_i to c_j in UMLS, where $l_{c_i.c_j} = 0$ *iff* $c_i = c_j$, or equivalently, *iff* the two concepts are identical. We choose the isa relation directed from the concepts of d to the concepts of q, i.e. a query about '*animal*' will be satisfied by a document about '*dog*' but not the inverse. Choosing one direction of the isa relation, namely from d to q, reduces the chance of adding noisy concepts to d.

Concerning the semantic similarity measure $Sim_{isa}(c_i, c_j)$, or simply $Sim(c_i, c_j)$, there are several possible choices [Aslam & Frost, 2003; Holi & Hyvönen, 2005; Leacock & Chodorow, 1998; Li *et al.*, 2003; Mohler & Mihalcea, 2009]. The main constraint is to be monotonically decreasing with respect to the length of the isa-path between the two concepts $l_{c_i.c_j}$. We choose the *exponential* function to estimate the semantic similarity between two concepts that are related via an isa relation:

$$Sim(c_i, c_j) = \begin{cases} 0 & \text{no semantic relation} \\ e^{-\beta \times l_{c_i.c_j}} & c_i \xrightarrow{isa} c_j \end{cases} \tag{7.1}$$

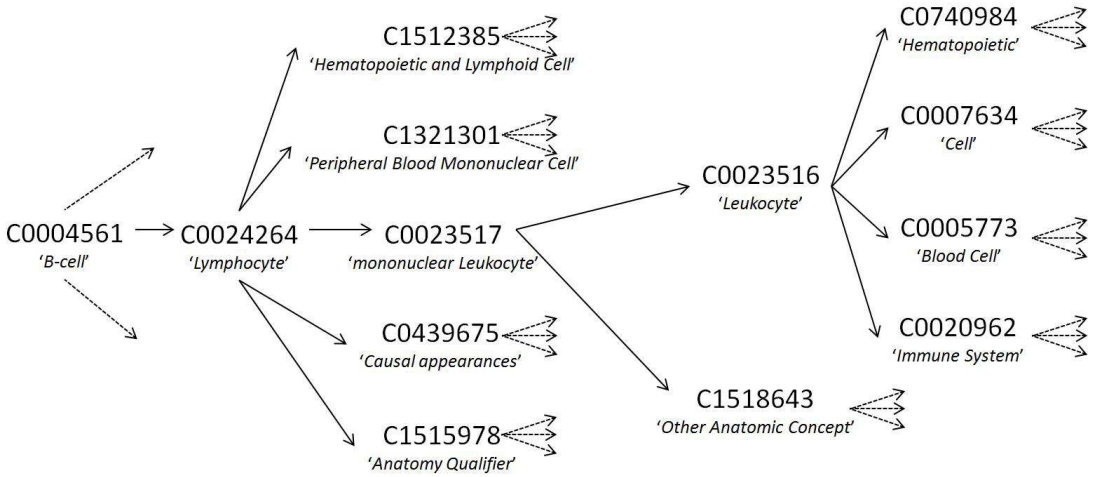
where $\beta \in \mathbb{R}^{+*}$ is a tuning parameter. Figure 7.2 shows the exponential semantic similarity function with different values of β.

Figure 7.2: The exponential semantic similarity measure



One possible justification of choosing exponential measure, besides its good experimental results [Abdulahhad *et al.*, 2011c], is that it rapidly decreases. Actually, this is a very important property in any similarity measure, because concepts rapidly become very general and far from the original concept. For example, (Figure 7.3) shows the isa-paths starting from the concept '*C0004561*' that corresponds to '*B-cell*'. We can see that concepts rapidly, within 2 or 3 edges, become very general, and have very far meaning comparing to the meaning of the original concept '*C0004561*'.

Figure 7.3: The isa-paths starting from '*B-cell*' in UMLS



7.3 Term Weighting

Whatever the type of indexing terms is, either words or concepts, term weighting is semi-mandatory to build effective and high-performance IR models. Fang et al. [Fang *et al.*, 2004] review, in a form of constraints, the weighting heuristics that weighting functions should respect in order to be effective. Table 7.1 lists the constraints of Fang et al. and the intuitions beyond. Clinchant et al. [Clinchant & Gaussier, 2010] presents an analytical form of these constraints.

In this study, we test three instances of our model: Exhaustivity and Specificity (ES) instance (Equation 6.2–P.107), relation-based (RL) instance (Equation 6.6–P.112), and structure-based (ST) instance (Equation 6.7–P.115). Each instance contains many weighting components that need to be concretely defined. In the following subsections, we present a detailed definitions of all these components, to finally our instances can be experimentally tested. As we mentioned earlier in (Chapter 6–P.101), the instances ES & RL use classical weighting mechanisms. However, the instance ST uses a different weighting mechanism that reflects the internal structure of phrases within text (Appendix A–P.165).

7.3.1 Classical Weighting

One of the most important weights is the weight of a term a_i in a document d, denoted w_i^d. In this study and since we depend on the vector space framework to build our instances, we choose a variant of the well-known *tf.idf* weighting schema [Luhn, 1958]. Actually, we use (Equation 7.2), which respects the first four constraints *TFC1 & TFC2 & TDC & LNC1* of (Table 7.1) [Fang & Zhai, 2005].

$$w_i^d = \frac{c(a_i, d)}{c(a_i, d) + \frac{|d|}{avdl}} \times \frac{N}{df(a_i)} \tag{7.2}$$

where, $c(a_i, d)$ is the count, or the number of occurrences, of a_i in d, $|d|$ is the document length, $avdl$ is the average document length in the corpus, N is the total number of documents in the corpus, and $df(a_i)$ is the number of documents in the corpus that contains a_i.

Table 7.1: Term weighting constraints

Constraints	**Intuitions**
TFC1	to favor a document with more occurrence of a query term
TFC2	to favor document matching more distinct query terms
TFC2	to make sure that the change in the score caused by increasing TF (Term Frequency) from 1 to 2 is larger than that caused by increasing TF from 100 to 101
TDC	to regulate the impact of TF and IDF (Inverse Document Frequency): it ensures that, given a fixed number of occurrences of query terms, we should favor a document that has more occurrences of discriminative terms (i.e. high IDF terms)
LNC1	to penalize a long document
LNC2, TF-LNC	to avoid over-penalize a long document: as it says that if we concatenate a document with itself k times to form a new document, then the score of the new document should not be lower than the original document
TF-LNC	to regulate the interaction of TF and document length: if d_1 is generated by adding more occurrences of the query term to d_2, the score of d_1 should be higher than d_2

In ES instance, the weight of the term a_i, which is either a word or a concept, in a document d is defined as follows:

$$w_i^d = \begin{cases} \frac{c(a_i,d)}{c(a_i,d)+\frac{|d|}{avdl}} \times \frac{N}{df(a_i)} & \text{If} \quad a_i \in \Omega_d \\ 0 & \text{otherwise} \end{cases} \tag{7.3}$$

In RL instance, the weight of the term a_i, which is a concept, in the original document d^o is defined as follows:

$$w_i^{d^o} = \begin{cases} \frac{c(a_i,d^o)}{c(a_i,d^o)+\frac{|d^o|}{avdl}} \times \frac{N}{df(a_i)} & \text{If} \quad a_i \in \Omega_d^o \\ 0 & \text{otherwise} \end{cases} \tag{7.4}$$

We assume that query's terms are equally important. We thus define $w_i^q = \frac{1}{|q|}$, where $|q|$ is the query length. In both ES & RL instances, the weight of the term a_i in a query q is defined as follows:

$$w_i^q = \begin{cases} \frac{1}{|q|} & \text{If} \quad a_i \in \Omega_q \\ 0 & \text{otherwise} \end{cases} \tag{7.5}$$

In IR, documents are ranked with respect to a query and not the inverse. In addition, the document length could change in the RL instance, where $|d^o| \leq |d^r|$, whereas the query length remains unchanged. We thus decide to weight the terms of the meet object $\mu(d) \curlywedge \mu(q)$ using the query, where $w_i^{dq} = c(a_i, q)$ and $c(a_i, q)$ is the count of the term a_i in the query q.

In ES instance, the weight of the term a_i, which is either a word or a concept, in the meet object $\mu(d) \curlywedge \mu(q)$ is defined as follows:

$$w_i^{dq} = \begin{cases} c(a_i, q) & \text{If} \quad a_i \in \Omega_d \cap \Omega_q \\ 0 & \text{otherwise} \end{cases} \tag{7.6}$$

In RL instance, the weight of the term a_i, which is a concept, in the meet object of the original document $\mu(d^o) \curlywedge \mu(q)$ is defined as follows:

$$w_i^{d^o q} = \begin{cases} c(a_i, q) & \text{If} \quad a_i \in \Omega_d^o \cap \Omega_q \\ 0 & \text{otherwise} \end{cases} \tag{7.7}$$

In RL instance also, the weight of the term a_i, which is a concept, in the meet object of the expanded document $\mu(d^r) \curlywedge \mu(q)$ is defined as follows:

$$w_i^{d^r q} = \begin{cases} c(a_i, q) & \text{If} \quad a_i \in \Omega_d^r \cap \Omega_q \\ 0 & \text{otherwise} \end{cases} \tag{7.8}$$

After these definitions, namely (Equations 7.1&7.3&7.4&7.5&7.6&7.7&7.8), the two instances ES & RL are ready to be tested and experimentally studied. In the next subsection, we use the Relative Concept Frequency (RCF) approach (Appendix A–P.165), which reflects the internal structure of phrases, in order to define the weighting components of the ST instance.

7.3.2 Relative Weighting

We show in (Appendix A–P.165), our new counting approach RCF that takes the internal phrase structure into account. RCF extends the flat representation of documents and queries through exploiting some structural relations between terms. Therefore, we argue that this approach is suitable for the ST instance of our model (Equation 6.7–P.115). Since our new approach of counting maintains the document and query length, then the only component in (Equations 7.2&7.5&7.6) that must be changed is the count $c(a, d)$.

The ST instance is similar to the ES instance. Therefore, in ST instance, the weight of the term a_i, which is a concept, in a document d is defined as follows:

$$w_i^d = \begin{cases} \frac{rcf(a_i,d)}{rcf(a_i,d)+\frac{|d|}{avdl}} \times \frac{N}{df(a_i)} & \text{If} \quad a_i \in \Omega_d \\ 0 & \text{otherwise} \end{cases} \tag{7.9}$$

where $rcf(a_i, d)$ is the relative frequency of a_i in d (Equation A.1–P.172). For more details about the way of computing $rcf(a_i, d)$, see (Appendix A–P.165). We assume that query's terms are equally important. We thus define $w_i^q = \frac{1}{|q|}$, where $|q|$ is the query length. In ST instance, the weight of the term a_i, which is a concept, in a query q is defined as follows:

$$w_i^q = \begin{cases} \frac{1}{|q|} & \text{If} \quad a_i \in \Omega_q \\ 0 & \text{otherwise} \end{cases} \tag{7.10}$$

In ST instance, the weight of the term a_i, which is a concept, in the meet object $\mu(d) \curlywedge \mu(q)$ is defined as follows:

$$w_i^{dq} = \begin{cases} rcf(a_i, q) & \text{If} \quad a_i \in \Omega_d \cap \Omega_q \\ 0 & \text{otherwise} \end{cases} \tag{7.11}$$

After these definitions, namely (Equations 7.9&7.10&7.11), the instance ST is ready to be tested and experimentally studied.

7.4 Test Collections (Corpora)

In order to study the retrieval performance of the instances of our model, we apply them to *six* corpora: *four* from *ImageCLEF*[1], and *two* from *TREC*[2].

7.4.1 ImageCLEF

ImageCLEF is a part of CLEF[3] (Cross-Language Evaluation Forum), which is a yearly campaign for evaluation of multilingual information retrieval since 2000. For example, ImageCLEF2012 contains four main tracks: 1) Medical Image Classification and Retrieval, 2) Photo Annotation and Retrieval, 3) Plant Identification, and 4) Robot Vision. Medical Image Classification and Retrieval track contains three tasks: a) modality classification, b) ad-hoc image-based retrieval which is an image retrieval task using textual, image or mixed queries, and c) case-based retrieval: in this task the documents are journal articles extracted from PubMed[4] and the queries are case descriptions. In this study, we only consider ad-hoc image-based corpora, denoted $clef$. In addition, we only use the textual part of corpora. Furthermore, since corpora contain medical data, we index documents and queries using the two types of indexing terms, namely words and concepts.

$clef09$, $clef10$, $clef11$, $clef12$: are four ad-hoc image-based corpora of the years 2009, 2010, 2011, and 2012, respectively. These corpora contain *short* medical documents and queries. Figures 7.4&7.5 show an example of a query and a document from the $clef09$ corpus, respectively. What we exactly index in queries is their English parts `EN_DESCRIPTION`. Whereas, we index the `caption` and the `title` in documents.

Figure 7.4: An example of a query in $clef09$

```
<topic>
  <ID>2</ID>
  <TYPE>visual</TYPE>
  <EN_DESCRIPTION>Breast cancer mammogram</EN_DESCRIPTION>
  <FR_DESCRIPTION>Mammographies d'un cancer du sein</FR_DESCRIPTION>
  <DE_DESCRIPTION>Mammogramm mit Brustkrebs</DE_DESCRIPTION>
</topic>
```

Table 7.2 shows some statistics about the medical corpora of ImageCLEF using two types of terms: words and concepts, where the statistics of words are taken after removing stop words and stemming, and $avdl, avql$ are the average length of documents and queries respectively. Furthermore, (Table 7.2) clearly shows a bias in the length of documents and queries between the word-space and the concept-space. Actually, in view that mapping tools map each phrase to a set of candidate concepts, then documents and queries in the concept space are much longer, e.g. a phrase like '*xray*' is mapped to six different UMLS concepts using MetaMap.

[1] www.imageclef.org
[2] trec.nist.gov
[3] www.clef-campaign.org
[4] www.ncbi.nlm.nih.gov/pubmed

Figure 7.5: An example of a document in $clef09$

```
<record>
  <figureID>27982</figureID>
  <figureURL>http://radiology.rsnajnls.org/cgi/content/full/210/1/11/F4A</figureURL>
  <caption>Figure 4a. Disappearance of the thymus after irradiation in an 8-week-old infant
      with "symptoms of thymic enlargement," according to the original caption. (a) Arrows in
      an initial frontal radiograph point out "atelectasis" of the right lung and
      accompanying thymic enlargement, an appearance that today is recognized as a normal
      finding due to the thymus gland. (b) Radiograph obtained 5 weeks after thymic
      irradiation (dose not given) shows that the thymus is notably smaller; the original
      noted that the child "is now enjoying perfect health." (Reprinted, with permission,
      from reference 28.)</caption>
  <title>The right place at the wrong time: historical perspective of the relation of the
      thymus gland and pediatric radiology</title>
  <pmid>9885579</pmid>
  <articleURL>http://radiology.rsnajnls.org/cgi/content/full/210/1/11</articleURL>
  <imageLocalName>27982.jpg</imageLocalName>
</record>
```

Table 7.2: Statistics of ImageCLEF corpora

Corpus	#d	#q	Terms	$avdl$	$avql$
$clef09$	74901	25	Words	62.16	3.36
			Concepts	157.48	10.84
$clef10$	77495	16	Words	62.12	3.81
			Concepts	157.27	12.0
$clef11$	230088	30	Words	44.83	4.0
			Concepts	101.92	12.73
$clef12$	306530	22	Words	47.16	3.55
			Concepts	47.16	9.41

7.4.2 TREC

TREC (Text REtrieval Conference) was started in 1992. It supports research within the IR community by providing the infrastructure necessary for large-scale evaluation of text retrieval methodologies. TREC encompasses many different tracks[1]. In this study, we only consider two corpora $trec6$ and $trec8$ of the ad-hoc retrieval task. The two corpora are only indexed by words.

$trec6$**,** $trec8$: contain *long* general-content documents and queries. Figure 7.6 shows the query number 301 from $trec6$. Figure 7.7 shows what a document from $trec$ looks like. For TREC queries, we use the three fields: `title`, `desc`, and `narr`. Concerning $trec6$, documents on disks 4 & 5 and topics 301-350 are used. Concerning $trec8$, documents on disks 4 & 5 without Congressional Record (CR) and topics 401-450 are used.

Table 7.3 shows some statistics about the two corpora $trec6$ and $trec8$ that are indexed using words, where the statistics of words are taken after removing stop words and stemming, and $avdl, avql$ are the average length of documents and queries respectively.

[1] trec.nist.gov/tracks.html

Figure 7.6: The query 301 of *trec*6

```
<top>
  <num> Number: 301
  <title> International Organized Crime
  <desc> Description:
         Identify organizations that participate in international criminal activity, the activity,
         and, if possible, collaborating organizations and the countries involved.
  <narr> Narrative:
         A relevant document must as a minimum identify the organization and the type of illegal
         activity (e.g., Columbian cartel exporting cocaine). Vague references to international drug
         trade without identification of the organization(s) involved would not be relevant.
</top>
```

Figure 7.7: The document structure in *trec*

```
<DOC>
  <DOCNO>...</DOCNO>
  <TEXT>...</TEXT>
</DOC>
```

Table 7.4 shows an overview about the corpora that are used in this thesis and about their content. We tried to apply the instances of our model to corpora of different fields (medical vs. general) and with a variety of documents and queries length. This diversity gives more credibility to our experiments.

7.5 Metrics & Tools

In order to compare the retrieval performance of IR models, we use the Mean Average Precision (MAP) metric, which is both recall and precision metric, and also the precision at the first ten documents (P@10) metric, which is a pure precision metric (Equations II.31 & II.35 in [Dinh, 2012] respectively).

The statistical significance tests are used to verify if a system a is statistically better than another system b, and that it is not the pure coincidence that makes a better than b. As statistical significance test, we use Fisher's Randomization test at the 0.05 level [Smucker *et al.*, 2007].

Besides MetaMap, which is used to map text to UMLS concepts, and `trec_eval`[1], which is used to compute the MAP and P@10 metrics, we build our own tools, using Java, to:

- extract the word-based index of documents and queries.
- compute the Relative Concept Frequency (see Section A.2.1–P.167), which is used to weight the concepts in the ST instance of our model.
- build an IR system, which applies retrieval formulae to corpora.

[1] trec.nist.gov/trec_eval/

Table 7.3: Statistics of *trec*6 and *trec*8 corpora

Corpus	#d	#q	*avdl*	*avql*
*trec*6	551787	50	266.67	46.56
*trec*8	523865	50	243.87	29.06

Table 7.4: Corpora overview

Corpus	Terms	Description
*clef*09	Words & Concepts	Contain *short* medical documents and queries
*clef*10		
*clef*11		
*clef*12		
*trec*6	Words	Contain *long* general-content documents and queries
*trec*8		

7.6 Baselines

In order to check the performance of our model, we must compare it to some high-performance baselines. In order to obtain more valuable and reliable results, we choose baseline models belonging to different mathematical frameworks:

Vector Space. From the vector space models family, we choose the Pivoted Normalization Method [Singhal *et al.*, 1996], denoted piv (Equation 7.12), where s is a tuning parameter, and usually $s = 0.2$ [Fang *et al.*, 2004; Singhal, 2001; Singhal *et al.*, 1996; Zhai, 2008].

$$RSV_{piv}(d,q) = \sum_{a \in d \cap q} \frac{1 + \ln\left(1 + \ln\left(c(a,d)\right)\right)}{(1-s) + s\frac{|d|}{avdl}} \times c(a,q) \times \ln \frac{N+1}{df(a)} \tag{7.12}$$

Probabilistic Models. From the probabilistic IR models family, we choose the BM25 model [Robertson & Walker, 1994], denoted $bm25$ (Equation 7.13), where b, k_1, k_3 are tuning parameters, and usually $b = 0.75$, $k_1 = 1.2$, and $k_3 = 1000$ [Fang & Zhai, 2005; Fang *et al.*, 2004; Robertson & Walker, 1994].

$$RSV_{bm25}(d,q) = \sum_{a \in d \cap q} \ln \frac{N - df(a) + 0.5}{df(a) + 0.5} \times \frac{(k_1+1) \times c(a,d)}{k_1 \times \left((1-b) + b \times \frac{|d|}{avdl}\right) + c(a,d)} \times \frac{(k_3+1) \times c(a,q)}{k_3 + c(a,q)} \tag{7.13}$$

Language Models. From the language models family [Ponte & Croft, 1998], we choose two smoothing methods. First, Dirichlet priors method, denoted dir (Equation 7.14), where $p(a, D)$ is the probability of a given the corpus language model D, μ is a tuning parameter, and usually $\mu = 2000$ [Fang *et al.*, 2004; Zhai & Lafferty, 2001]. Second, Jelinek-Mercer method, denoted jm (Equation 7.15), where λ is a tuning parameter, and usually $\lambda = 0.1$ for short queries, which is the case in *clef* corpora. Whereas, $\lambda = 0.7$ for long queries, which is the case in *trec* corpora

[Zhai & Lafferty, 2001].

$$RSV_{dir}(d,q) = |q| \times \ln \frac{\mu}{|d|+\mu} + \sum_{a \in d \cap q} c(a,q) \times \ln \left(1 + \frac{c(a,d)}{\mu \times p\,(a,D)}\right) \tag{7.14}$$

$$RSV_{jm}(d,q) = |q| \times \ln(\lambda) + \sum_{a \in d \cap q} c(a,q) \times \ln \left(1 + \frac{1-\lambda}{\lambda} \times \frac{c(a,d)}{|d| \times p\,(a,D)}\right) \tag{7.15}$$

Information-Based Models. From the information-based models family [Clinchant & Gaussier, 2010], we choose Log-Logistic distribution, denoted *lgd* (Equation 7.16), where γ is a tuning parameter, and usually $\gamma = 1$ [Amati & Van Rijsbergen, 2002].

$$RSV_{lgd}(d,q) = \sum_{a \in d \cap q} -c(a,q) \times \log \left(\frac{\frac{df(a)}{N}}{c(a,d) \times \log\left(1 + \gamma \frac{avdl}{|d|}\right) + \frac{df(a)}{N}}\right) \tag{7.16}$$

Table 7.5 shows a global overview of all baseline models used and their tuning parameter values. As we can see from the table, we choose baseline models belonging to a variety of mathematical frameworks.

Table 7.5: Overview of baseline models

Model	Framework	Equation	Parameters
piv	vector space models	Equation 7.12	$s = 0.2$
bm25	probabilistic models	Equation 7.13	$k_1 = 1.2$
			$k_3 = 1000$
			$b = 0.75$
dir	language models	Equation 7.14	$\mu = 2000$
jm	language models	Equation 7.15	$\lambda = 0.1$ (short queries)
			$\lambda = 0.7$ (long queries)
lgd	information models	Equation 7.16	$c = 1$

7.6.1 Results

Table 7.6 shows the MAP and P@10 of all baselines, where we apply the baseline models to all corpora after indexing documents and queries using words.

Table 7.7 shows the MAP and P@10 of all baselines, where we apply the baseline models to the corpora of ImageCLEF after mapping the text of documents and queries to UMLS concepts via MetaMap.

7.7 Conclusion

We presented in this chapter all technical details that are necessary to test and experiment the instances of our model.

Table 7.6: Baselines using words

Corpus	Metric	*piv*	*bm25*	*dir*	*jm*	*lgd*
*clef*09	MAP	0.3664	0.3726	0.3353	0.3792	0.3917
	P@10	0.5920	0.5800	0.5600	0.6040	0.6080
*clef*10	MAP	0.2992	0.2745	0.2960	0.2994	0.3106
	P@10	0.4312	0.3187	0.4250	0.3875	0.4312
*clef*11	MAP	0.1546	0.1995	0.1534	0.1985	0.1960
	P@10	0.3033	0.3367	0.2433	0.3167	0.3267
*clef*12	MAP	0.1027	0.1438	0.1161	0.1371	0.1420
	P@10	0.2182	0.2682	0.2182	0.2818	0.3000
*trec*6	MAP	0.2076	0.2238	0.2410	0.2532	0.2064
	P@10	0.4340	0.4320	0.4100	0.4460	0.3880
*trec*8	MAP	0.2302	0.2521	0.2514	0.2627	0.2318
	P@10	0.4640	0.4760	0.4360	0.4820	0.4380

Table 7.7: Baselines using concepts

Corpus	Metric	*piv*	*bm25*	*dir*	*jm*	*lgd*
*clef*09	MAP	0.2626	0.2672	0.2675	0.3058	0.2966
	P@10	0.4440	0.4600	0.4640	0.5280	0.5080
*clef*10	MAP	0.2530	0.2127	0.2455	0.2451	0.2525
	P@10	0.3687	0.2937	0.3625	0.3750	0.3937
*clef*11	MAP	0.1096	0.1552	0.1228	0.1580	0.1512
	P@10	0.2300	0.3100	0.2333	0.2800	0.2833
*clef*12	MAP	0.0934	0.1034	0.0861	0.1022	0.1063
	P@10	0.1318	0.1500	0.1364	0.1591	0.1727

We extract the list of words that index documents and queries after removing stop words and stemming using Porter algorithm. In addition, we use MetaMap to map the textual content of documents and queries to UMLS concepts, and to finally build the conceptual index of documents and queries.

Since we use the vector space framework to implement our instances, we weight indexing terms, either words or concepts, using a variant of the classical *tf.idf* weighting schema. We also use a new concept counting approach, namely RCF, where RCF takes the internal hierarchical structure of phrases into account.

We apply our model to six corpora: four from ImageCLEF and two from TREC. We use the MAP and P@10 metrics for comparing the retrieval performance of our model with the performance of some high-performance baseline models. Actually, these baselines belong to a variety of mathematical frameworks, and that gives more credibility for the comparison. To check if one model is statistically better than another one, we use the Fisher's Randomization test at the 0.05 level.

We use a variety of corpora and baselines in order to obtain more credible and useful conclusions.

Chapter 8

Results and Discussion

8.1 Introduction

After presenting all necessary technical details in (Chapter 7–P.119), we present in this chapter the experimental results of the instances of our model. We also compare the retrieval performance of our model with the performance of the baseline models, in order to finally draw experimental conclusions about our model.

By presenting these results of the instances of our model, namely ES, RL, ST instances (Chapter 6–P.101), our main goal is, on the one hand, to show the applicability of our model to large-scale corpora, and on the other hand, to place our model with respect to other IR models.

Concerning each instance alone, the main purpose of the experiments that we manage on the ES instance is to show the importance of integrating both Exhaustivity and Specificity in one IR model. Concerning the RL instance, the goal is to show the flexibility and the ability of our model to formally and efficiently integrate large knowledge resources into an IR model. Our main intention beyond testing the ST instance is to show the validity of hypotheses that govern our new structure-based concept counting approach (Appendix A–P.165).

The chapter is organized as follows: in section 2, we present, in view of the choices that we made in (Chapter 7–P.119), the retrieval formulae of the instances of our model, namely the ES, RL, and ST instances. Section 3 shows the experimental results of applying the ES instance to the selected corpora. In sections 4 & 5, we present the experimental results of the RL and ST instances, respectively. We conclude in section 6.

8.2 Retrieval Formulae

Based on the weighting choices that we made in the previous chapter, namely (Equations 7.3 & 7.5 & 7.6), the retrieval formula (Equation 6.2–P.107) of the ES instance can be rewritten as follows[1]:

$$\begin{aligned} RSV_{ES}(d,q) \propto \quad & \alpha \times \log\left(\sum_{a_i \in d \cap q} c(a_i,q)\right) \\ & +(1-\alpha) \times \log\left(\sum_{a_i \in d \cap q} c(a_i,q) \times \frac{c(a_i,d)}{c(a_i,d)+\frac{|d|}{avdl}} \times \frac{N}{df(a_i)}\right) \end{aligned} \tag{8.1}$$

[1]For simplifying the notation, we replace Ω_d, Ω_q, and $\Omega_d \cap \Omega_q$, by d, q, and $d \cap q$, respectively.

In the same manner, and based on the weighting choices that we made and the semantic similarity measure that we considered in the previous chapter, namely (Equations 7.1 & 7.4 & 7.5 & 7.7 & 7.8), we can rewrite the retrieval formula (Equation 6.6–P.112) of the RL instance, as follows:

$$RSV_{RL}(d,q) = \left(\sum_{a_i \in d^o \cap q} c(a_i,q)\right) \times \left(\sum_{a_i \in d^r \cap q} c(a_i,q)\right) \times \left(\begin{array}{c} \sum_{a_i \in d^o \cap q} c(a_i,q) \times \frac{c(a_i,d^o)}{c(a_i,d^o)+\frac{|d^o|}{avdl}} \times \frac{N}{df(a_i)} \\ + \\ \sum_{a_i \in d^r \cap q} c(a_i,q) \times e^{-\beta \times l_{a_j \cdot a_i}} \times \frac{c(a_j,d^o)}{c(a_j,d^o)+\frac{|d^o|}{avdl}} \times \frac{N}{df(a_j)} \end{array} \right) \tag{8.2}$$

The retrieval formula (Equation 6.7–P.115) of the ST instance is very similar to the formula of the ES instance. The only difference is that, instead of the classical term-frequency, we use the relative concept frequency. Therefore, based on the weighting choices that we made in the previous chapter, namely (Equations 7.9 & 7.10 & 7.11), the retrieval formula of the ST instance can be rewritten as follows:

$$\begin{aligned} RSV_{ST}(d,q) \propto \quad & \alpha \times \log\left(\sum_{a_i \in d \cap q} rcf(a_i,q)\right) \\ & +(1-\alpha) \times \log\left(\sum_{a_i \in d \cap q} rcf(a_i,q) \times \frac{rcf(a_i,d)}{rcf(a_i,d)+\frac{|d|}{avdl}} \times \frac{N}{df(a_i)}\right) \end{aligned} \tag{8.3}$$

8.3 The ES Instance

The main goal of testing this instance, namely ES, is to show the importance of Exhaustivity and Specificity if they are appropriately defined and used. In other words, we show that integrating Exhaustivity and Specificity into one IR model can increase the effectiveness of that model. Although, this hypothesis is not new, but the new thing is the new practical definitions of Exhaustivity and Specificity presented in this thesis (Section 5.4.2–P.90).

We divide our experiments into three main categories: 1- experiments to study the effect of α value on the retrieval performance of ES (Equation 8.1), in other words, to study the *mutual effect* between Exhaustivity and Specificity, 2- experiments using words to compare the retrieval performance of the ES instance with the performance of the baseline models, and 3- experiments using concepts to compare the performance of the ES instance with the performance of the baseline models.

8.3.1 The Mutual Effect between Exhaustivity & Specificity

In this category of experiments, we only use words as indexing terms. By varying the value of α, in the ES instance, between $\alpha = 0.0$ (only Specificity) and $\alpha = 1.0$ (only Exhaustivity), it is possible to study the mutual effect between Exhaustivity and Specificity.

In (Table 8.1), we apply the ES instance to all corpora using words, (*) means a significant improvement comparing to the case of $\alpha = 0.0$ '*only Specificity*' and (†) means a significant improvement comparing to the case of $\alpha = 1.0$ '*only Exhaustivity*'. In (Table 8.1), we also see that, on the one hand, integrating the two components (Exhaustivity and Specificity) is better than depending on only one. On the other hand, there should be a type of balance between the

Table 8.1: The *ES* instance (Exhaustivity vs. Specificity)

α	*clef*09	*clef*10	*clef*11	*clef*12	*trec*6	*trec*8
0.0 (Specificity)	0.3149	0.3117	0.1965	0.1207	0.2116	0.1934
0.1	0.3340 †∗	0.3266 †∗	0.2113 †∗	0.1376 †∗	0.2295 †∗	0.2166 †∗
0.2	0.3544 †∗	0.3352 †∗	0.2229 †∗	0.1477 †∗	0.2464 †∗	0.2339 †∗
0.3	0.3653 †∗	**0.3402** †∗	0.2284 †∗	0.1539 †∗	0.2584 †∗	0.2480 †∗
0.4	0.3737 †∗	0.3102 †	**0.2330** †∗	0.1580 †∗	0.2639 †∗	0.2609 †∗
0.5	0.3789 †∗	0.3171 †	0.2317 †∗	**0.1586** †∗	**0.2661** †∗	**0.2666** †∗
0.6	0.3834 †∗	0.3221 †	0.2282 †	0.1539 †	0.2613 †∗	0.2631 †∗
0.7	**0.3846** †∗	0.3194 †	0.2221 †	0.1485 †	0.2357 †∗	0.2478 †∗
0.8	**0.3846** †∗	0.3144 †	0.2114 †	0.1463 †	0.1742 †	0.2112 †
0.9	0.3731 †∗	0.3056 †	0.1922 †	0.1348 †	0.0617 †	0.1271 †
1.0 (Exhaustivity)	0.2300	0.1921	0.0749	0.0551	0.0120	0.0369

two components. However, the exact value of α is corpus-dependent. In general, $\alpha = 0.5$, which is also the average of α values in all corpora, gives good performance (Figure 8.1). In the rest of the thesis, we fix $\alpha = 0.5$ to give an equal importance to both Exhaustivity and Specificity.

8.3.2 Experiments Using Words

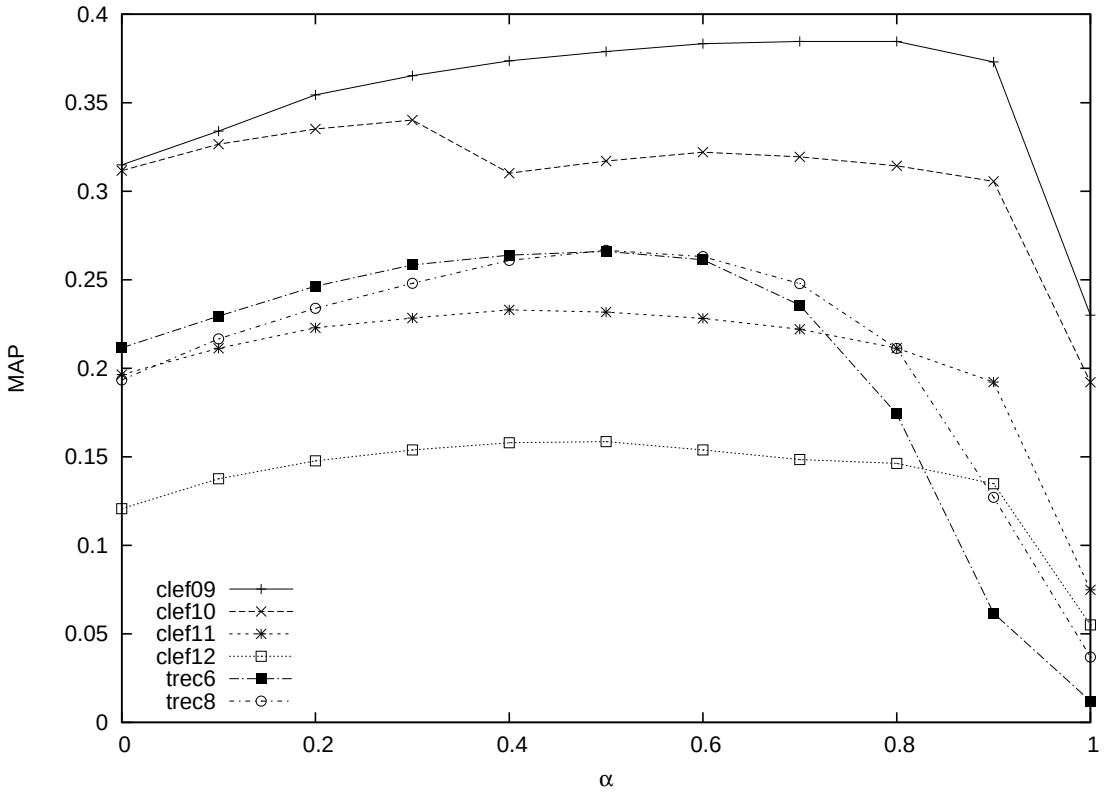
All experiments in this category are done using words as indexing terms. The main goal of this part of experiments is to study the retrieval performance of the *ES* instance of our model, and to compare it with the baseline models (Table 7.5–P.130).

In (Table 8.2), we apply the *ES* instance, *piv*, *bm25*, *jm*, *dir*, and *lgd* to all corpora using words, where (*) indicates that the *ES* instance is significantly better. Table 8.2 shows that the retrieval performance of the *ES* instance is better than the performance of some high-performance IR models, like *piv*, *bm25*, *jm*, *dir*, and *lgd*. That is especially clear when using a recall-precision metric like MAP. The only exception in (Table 8.2) is the *clef*09 corpus, where the *lgd* models performs better than other models including the *ES*. However, even though the MAP of *lgd* is higher than the MAP of *ES*, it is not statistically significant. Therefore, the *ES* instance and *lgd* have similar retrieval performance when applied to the *clef*09 corpus.

Anyway, the comparison using a pure-precision metric like P@10 shows that in three of six cases the *ES* instance performs better than other IR models. Similarly, even though *lgd* in *clef*09 and *jm* in *trec*6&*trec*8 perform better than *ES* but not in a statistically significant way. Therefore, it is possible to say that the *ES* instance performs better than or at least as good as the high-performance baseline IR models. Figure 8.2 shows precision at the standard recall levels.

Concerning the *trec*6 corpus, the performance of the *ES* instance is slightly better than the best run in TREC6 conference (the run '*anu6alol*' where $MAP = 0.2602$) [Voorhees & Harman, 2000], and it is also better than the best result obtained by the DFR framework (the run '$I(n_e)L2$' where $MAP = 0.2600$) [Amati & Van Rijsbergen, 2002]. Concerning the *trec*8 corpus, all runs in TREC8 conference, which have better score than our score, use some supplementary techniques. For example, query expansion, pseudo relevance feedback, POS tools,

Figure 8.1: The *ES* instance (Exhaustivity vs. Specificity)



and fusion of several runs [Voorhees & Harman, 1999].

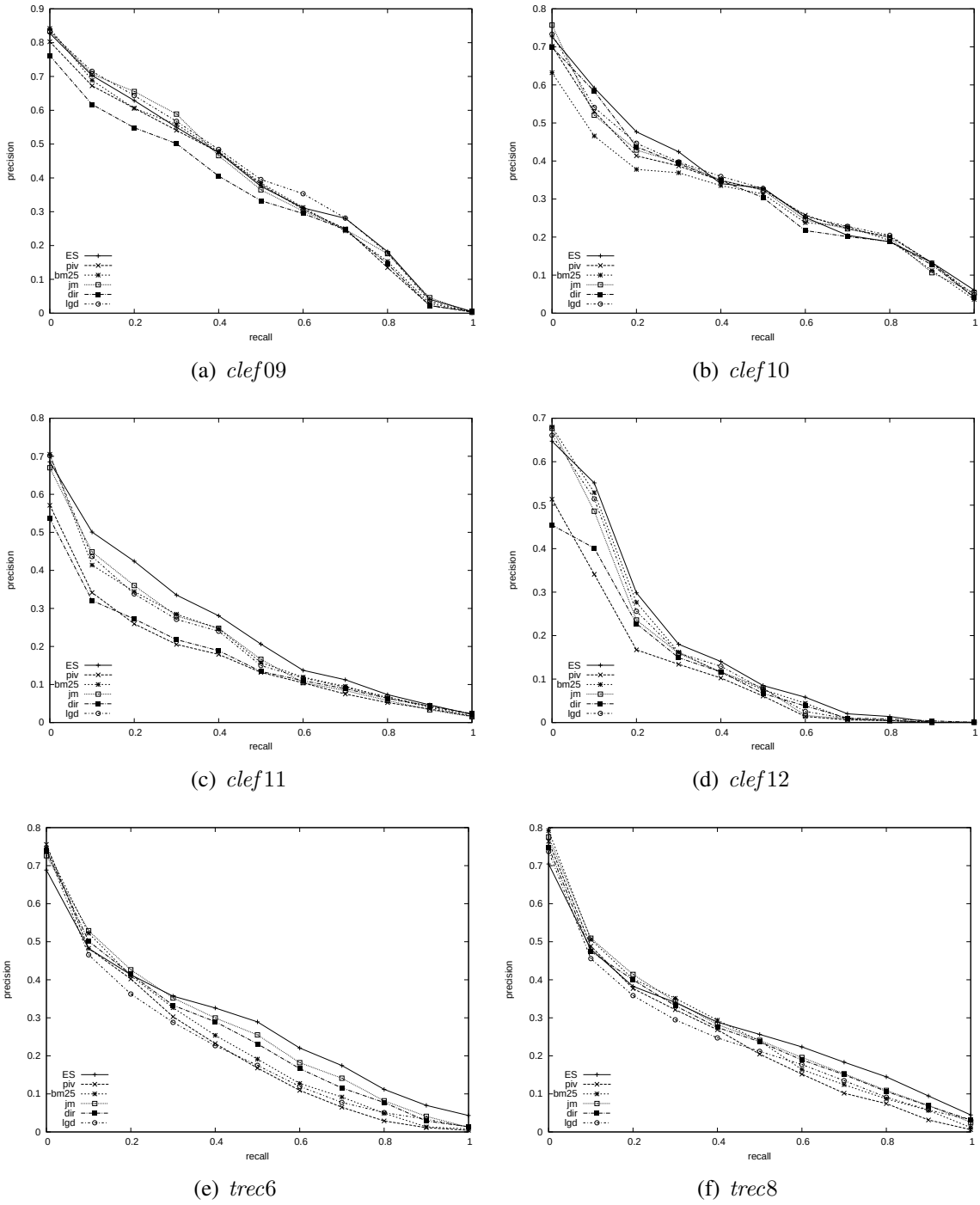
This category of experiments shows that, it is possible to build a high-performance IR model through exploiting the two notions: Exhaustivity and Specificity, or in other words, through the indirect matching between documents and queries. Table 8.2 shows the effect of integrating Exhaustivity and Specificity, namely the *ES* instance, on the retrieval performance. This type of results shows also that the indirect matching between d and q using an intermediate object $d \wedge q$ gives better results comparing to the direct matching.

8.3.3 Experiments Using Concepts

All experiments in this category are done using UMLS concepts as indexing terms. The main goal of this part of experiments is to study the retrieval performance of the *ES* instance of our model through repeating the same experiments of the previous section but using concepts instead of words. Furthermore, studying the experimental behavior of the *ES* instance using two different types of terms, namely words and concepts, gives more credibility to our conclusions and at the same time it shows the stability of the *ES* instance.

In (Table 8.3), we apply the *ES* instance, *piv*, *bm25*, *jm*, *dir*, and *lgd* to the corpora of ImageCLEF using concepts, where (*) indicates that the *ES* instance is significantly better. Table 8.3 shows that the retrieval performance of the *ES* instance is better than the performance of the baseline IR models.

Figure 8.2: The interpolated recall-precision using words in the ES instance



(a) $clef09$ (b) $clef10$

(c) $clef11$ (d) $clef12$

(e) $trec6$ (f) $trec8$

Table 8.2: The *ES* instance (experiments using words)

MAP	*clef*09	*clef*10	*clef*11	*clef*12	*trec*6	*trec*8
ES	0.3789	**0.3171**	**0.2317**	**0.1586**	**0.2661**	**0.2666**
piv	0.3664	0.2992	0.1546*	0.1027*	0.2076*	0.2302*
bm25	0.3726	0.2745*	0.1995*	0.1438	0.2238	0.2521
jm	0.3792	0.2994	0.1985*	0.1371	0.2532	0.2627
dir	0.3353*	0.2960	0.1534*	0.1161*	0.2410	0.2514
lgd	**0.3917**	0.3106	0.1960*	0.1420	0.2064*	0.2318*
P@10	*clef*09	*clef*10	*clef*11	*clef*12	*trec*6	*trec*8
ES	0.5960	**0.4500**	**0.3467**	**0.3455**	0.3680	0.4360
piv	0.5920	0.4312	0.3033	0.2182*	0.4340	0.4640
bm25	0.5800	0.3187*	0.3367	0.2682	0.4320	0.4760
jm	0.6040	0.3875	0.3167	0.2818	**0.4460**	**0.4820**
dir	0.5600	0.4250	0.2433*	0.2182*	0.4100	0.4360
lgd	**0.6080**	0.4312	0.3267	0.3000	0.3880	0.4380

The conclusions in this case, namely using concepts, are clearer, and in most cases our model outperforms other models. The only exception is that the P@10 of the *bm25* model is slightly better than the P@10 of *ES* in the *clef*11 corpus. Figure 8.3, which shows precision at the standard recall levels, also clarifies our conclusions about the outperformance of our model.

The results in (Table 8.3) show that integrating Exhaustivity and Specificity, through the *ES* instance, in an IR model can lead to important gain in the retrieval performance.

8.3.4 Discussion

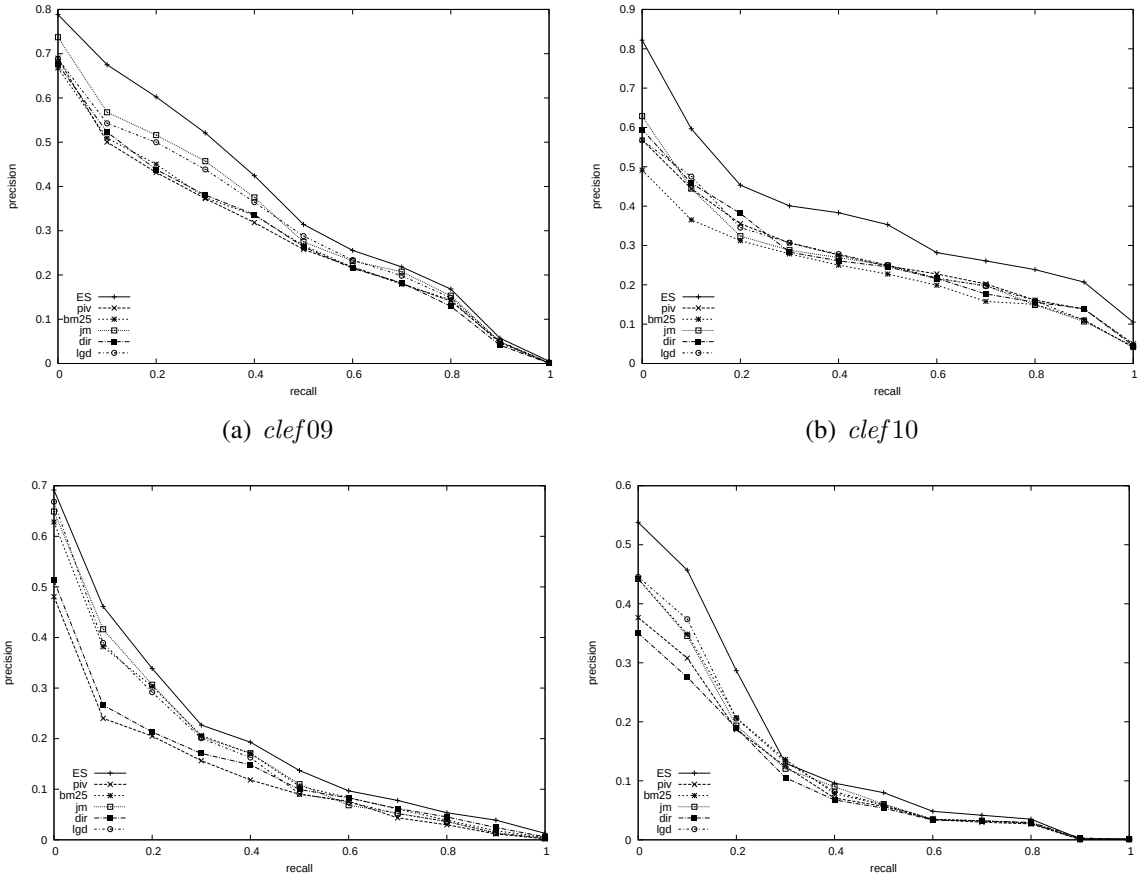
Comparing the results obtained using words (Table 8.2) with those of using concepts (Table 8.3) shows that indexing documents and queries using words generally gives better performance. We think that happens because natural languages are ambiguous. Therefore, mapping tools, like MetaMap, map each noun-phrase to several candidate concepts, and that adds a lot of noise. Actually, the text-concept mapping tools are very noisy. In addition, the classical IR models are built upon some statistical studies that concern words not concepts [Amati & Van Rijsbergen, 2002; Clinchant & Gaussier, 2010; Fang *et al.*, 2004; Luhn, 1958; Ponte & Croft, 1998; Robertson & Walker, 1994; Singhal *et al.*, 1996; Zhai & Lafferty, 2001], and mapping text to concepts, one way or another, destructs these statistical studies (see Appendix A–P.165). That's why classical IR models normally performs better when using words as indexing terms.

In general, experiments show that both Exhaustivity and Specificity are important to build a high-performance IR model, and there should be a type of balance between the two components. In addition, using two types of terms (words and concepts) gives more credibility to our obtained results, and shows that even though using words generally gives better results but using concepts gives more flexibility. That results from the available supplementary information that could be exploited besides concepts. This supplementary information is normally a part of the knowledge resources that contain concepts. Finally, the retrieval performance of our model (*ES* instance) has, in most cases, statistically significant improvement over the performance of the chosen high-performance baseline IR models.

Table 8.3: The ES instance (experiments using concepts)

MAP	$clef09$	$clef10$	$clef11$	$clef12$
ES	**0.3489**	**0.3448**	**0.1822**	**0.1322**
piv	0.2626*	0.2530	0.1096*	0.0934*
$bm25$	0.2672*	0.2127*	0.1552	0.1034
jm	0.3058	0.2451*	0.1580*	0.1022
dir	0.2675*	0.2455	0.1228*	0.0861*
lgd	0.2966*	0.2525*	0.1512*	0.1063
P@10	$clef09$	$clef10$	$clef11$	$clef12$
ES	**0.5760**	**0.4562**	0.3033	**0.2136**
piv	0.4440*	0.3687	0.2300	0.1318
$bm25$	0.4600*	0.2937*	**0.3100**	0.1500
jm	0.5280	0.3750*	0.2800	0.1591
dir	0.4640	0.3625	0.2333*	0.1364
lgd	0.5080	0.3937	0.2833	0.1727

Figure 8.3: The interpolated recall-precision using concepts in the ES instance



(a) $clef09$ (b) $clef10$

(c) $clef11$ (d) $clef12$

Table 8.4: The RL instance

MAP	$clef09$	$clef10$	$clef11$	$clef12$
RL	**0.3652**	0.3018	0.1797	0.1307
ES	0.3489*	**0.3448**	**0.1822**	**0.1322**
piv	0.2626*	0.2530	0.1096*	0.0934*
$bm25$	0.2672*	0.2127*	0.1552	0.1034*
jm	0.3058*	0.2451*	0.1580*	0.1022*
dir	0.2675*	0.2455	0.1228*	0.0861*
lgd	0.2966*	0.2525*	0.1512*	0.1063*
P@10	$clef09$	$clef10$	$clef11$	$clef12$
RL	**0.5760**	0.4375	0.2900	0.2091
ES	**0.5760**	**0.4562**	0.3033	**0.2136**
piv	0.4440*	0.3687	0.2300	0.1318
$bm25$	0.4600*	0.2937*	**0.3100**	0.1500
jm	0.5280	0.3750*	0.2800	0.1591
dir	0.4640	0.3625	0.2333	0.1364
lgd	0.5080	0.3937	0.2833	0.1727

8.4 The RL Instance

The main goal of testing the RL instance is to show the flexibility of our model. More precisely, we show the ability of our model to integrate large knowledge resources into IR models. Additionally, we still compare the experimental results of this instance with the baselines for placing our instance with respect to them.

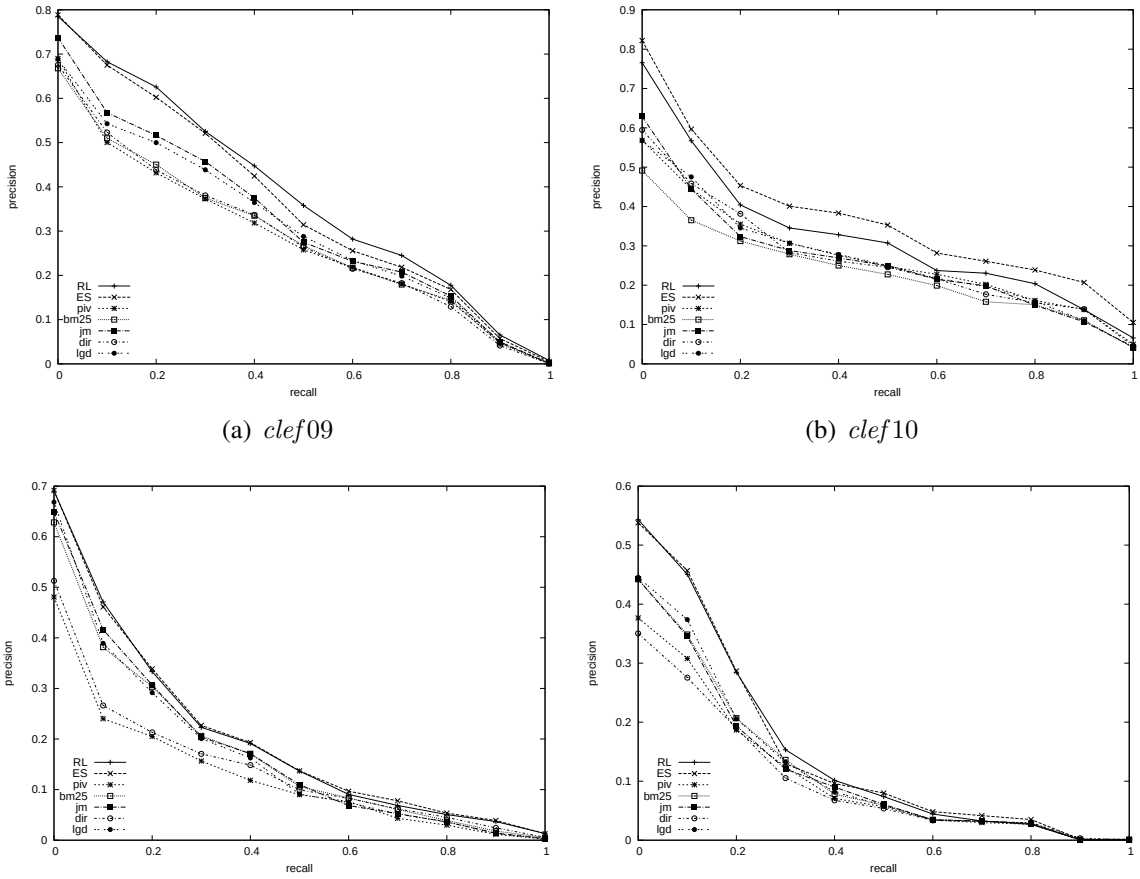
The RL instance exploits the semantic relations between concepts. Hence, we use concepts as indexing terms in the experiments of this instance. To test the hypothesis beyond this instance, we apply (Equation 8.2) to the ImageCLEF corpora, and then we compare the retrieval performance of RL with the performance of the ES instance, piv, $bm25$, jm, dir, and lgd models.

Table 8.4 shows the retrieval performance of applying RL, ES, piv, $bm25$, jm, dir, and lgd to the ImageCLEF corpora using concepts, where (*) indicates that the RL instance is significantly better. Concerning the tuning parameter β in the semantic similarity measure, we tested a range of values $\beta \in \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$ on all corpora, and we found that when $\beta = 7$ best results are obtained. Therefore, we fix $\beta = 7$ in all experiments of this instance.

In (Table 8.4), by comparing the retrieval performance of the RL instance, which exploits the isa relations between concepts, with the performance of the ES instance, which assumes that concepts are independent, we can see that RL performs as good as ES. Rather than the invalidity of the hypothesis beyond the RL instance, this result shows that more in-depth research is still needed in knowledge-based IR especially in the semantic similarity computation.

However, RL still outperforms other IR models (Table 8.4). Figure 8.4 also clarifies that. The most important case can be revealed by comparing the results of $clef09$ and $clef12$ in the two tables (Tables 8.3&8.4), where RL is always significantly better than other models, which is not the case in the ES instance.

Figure 8.4: The interpolated recall-precision in the RL instance



(a) $clef09$ (b) $clef10$

(c) $clef11$ (d) $clef12$

8.4.1 Discussion

The experiments of the RL instance reveal that we must be careful when exploiting semantic relations to expand documents. Even with a big value of β, which means a very small value of the semantic similarity measure, exploiting semantic relations adds some noise to documents. We can also see this noise by tracking the isa-paths starting from '*B-cell*' in UMLS (Figure 7.3–P.123). We can see that, in most cases, we rapidly get unrelated concepts, and very far from the original concept. In addition to this noise there is a type of redundancy in our instance, because the RL instance takes both the original document d^o and the expanded document d^r into account, where $d^o \subseteq d^r$.

Documents and queries in this instance are indexed by concepts. Therefore, on the one hand, moving from words to concepts increases the document and query lengths (Table 7.2), which means, queries and documents become longer and thus there is less need to expand them. On the other hand, concepts encompass synonymous words and phrases, and that means by using concepts we already exploit relations, and hence by using other types of relations the noise increases.

The performance of the RL instance presented in (Table 8.4) is for $\beta = 7$ in the semantic

Table 8.5: The ST instance

Model	$clef09$		$clef10$		$clef11$		$clef12$	
	MAP	Gain	MAP	Gain	MAP	Gain	MAP	Gain
ES	0.3488	+15%	0.3447	-19%	0.1822	+12%	0.1321	+7%
ST	0.4022*		0.2791		0.2031		0.1409	
	P@10	Gain	P@10	Gain	P@10	Gain	P@10	Gain
ES	0.5760	+9%	0.4562	-7%	0.3033	+7%	0.2136	+11%
ST	0.6280		0.4250		0.3233		0.2363	

similarity measure. We tested several values of $\beta \in \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$, on all corpora. The interesting thing is that the same value of β gives the best performance in all corpora. Hence, the value of β seems to be dependent on the knowledge resource that contains the concepts and relations, UMLS in our case, and not on corpora. In other words, it is preferable to optimize β on a knowledge resource and not on a particular corpus.

8.5 The ST Instance

As we mentioned, the ST instance (Equation 8.3) is very similar to the ES instance. Actually, the only difference between ST and ES is the term-frequency component, where in ES is classically computed, namely $c(a, d)$, whereas, in ST the Relative Concept Frequency (RCF) is used. RCF respects the internal structure of phrases in text (Appendix A–P.165).

Hence, what we exactly test in ST is the new way of term-frequency computing. In order to do that, we compare the retrieval performance of ST using RCF with its performance using the classical term-frequency, which is exactly the ES instance.

Table 8.5 shows the retrieval performance of applying ST and ES to the ImageCLEF corpora using concepts, where (*) indicates that the ST instance is significantly better. Table 8.5 shows that generally, in 3 corpora of 4, the new term-frequency computing method, namely RCF, improves the retrieval performance, and sometimes the improvement is statistically significant. Figure 8.5 shows precision at the standard levels of recall for the ST instance of our model.

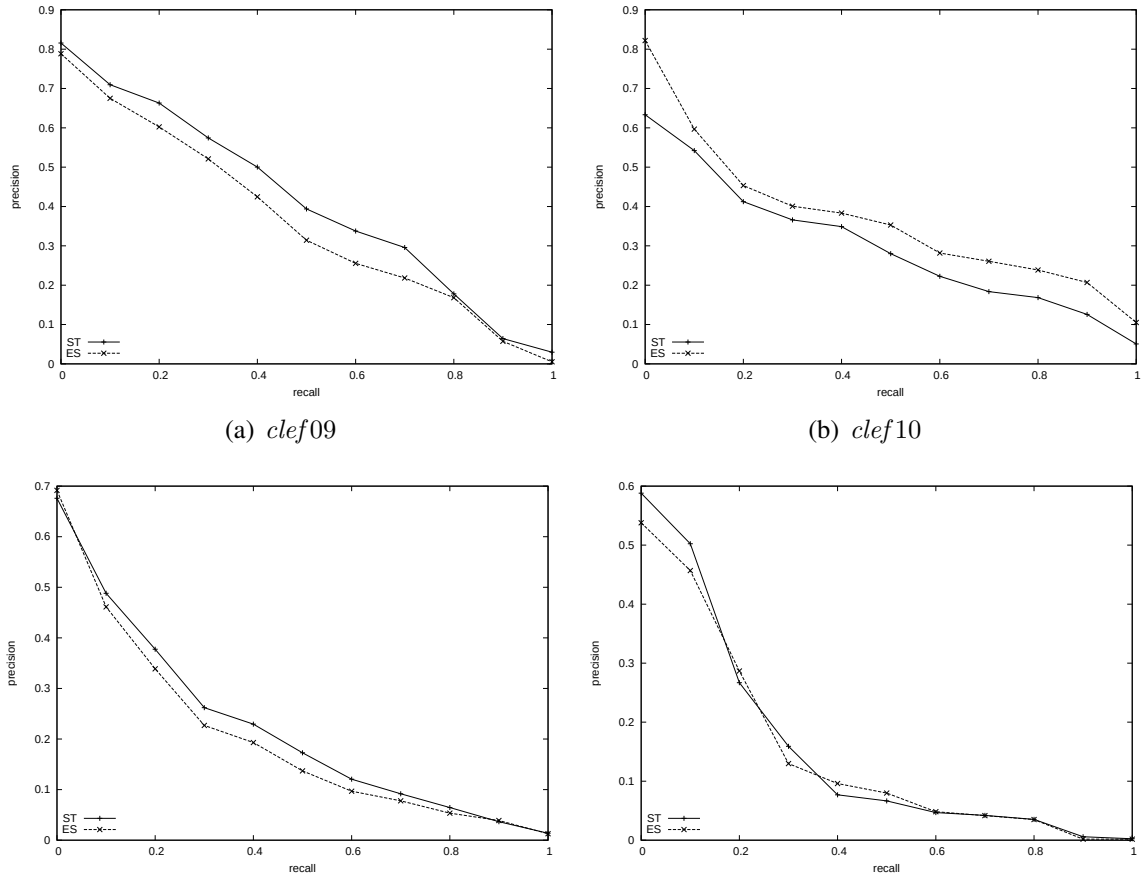
The interesting thing is that the RCF method can be integrated into the classical IR models [Abdulahhad *et al.*, 2013b], because this method maintains the document, query, and corpus lengths. Accordingly, we generalize our baseline models (Table 7.5–P.130) by replacing the classical term-frequency component $c(a, d)$ by the relative one $rcf(a, d)$ (Equation A.1–P.172).

Table 8.6 shows the retrieval performance of applying our baselines and their variants by using RCF to the ImageCLEF corpora using concepts, where (*) indicates that the variant of a baseline model using RCF is significantly better than its original form. Table 8.6 shows that in all cases, except when applying $bm25$ to the $clef11$ corpus, the new term-frequency computing method improves the retrieval performance, and in many times this improvement is statistically significant. For example, (Figure 8.6) compares the precision at the standard recall levels of the lgd model using classical term-frequency with the same model using the RCF.

Table 8.6: Integrating RCF in other IR models

Model	Weight	*clef*09		*clef*10		*clef*11		*clef*12	
		MAP	Gain	MAP	Gain	MAP	Gain	MAP	Gain
piv	classic	0.2626	+49%	0.2530	+14%	0.1096	+62%	0.0933	+46%
	RCF	0.3909*		0.2894		0.1780*		0.1360*	
bm25	classic	0.2672	+26%	0.2126	+20%	0.1552	-3%	0.1033	+11%
	RCF	0.3355		0.2558		0.1503		0.1142	
dir	classic	0.2675	+26%	0.2455	+12%	0.1227	+2%	0.0861	+13%
	RCF	0.3380		0.2748		0.1255		0.0969	
jm	classic	0.3057	+31%	0.2451	+14%	0.1579	+15%	0.1021	+29%
	RCF	0.4005*		0.2803		0.1820*		0.1318*	
lgd	classic	0.2965	+35%	0.2525	+13%	0.1512	+25%	0.1063	+28%
	RCF	0.3997*		0.2853		0.1887*		0.1365	
		P@10	Gain	P@10	Gain	P@10	Gain	P@10	Gain
piv	classic	0.4440	+39%	0.3687	+7%	0.2300	+45%	0.1318	+86%
	RCF	0.6160*		0.3937		0.3333*		0.2454*	
bm25	classic	0.4600	+17%	0.2937	+21%	0.3100	-6%	0.1500	+6%
	RCF	0.5400		0.3562		0.2900		0.1590	
dir	classic	0.4640	+24%	0.3625	+21%	0.2333	+1%	0.1363	+7%
	RCF	0.5760		0.4375		0.2366		0.1454	
jm	classic	0.5280	+16%	0.3750	+20%	0.2800	+20%	0.1590	+51%
	RCF	0.6120		0.4500*		0.3366		0.2409*	
lgd	classic	0.5080	+23%	0.3937	+13%	0.2833	+10%	0.1727	+34%
	RCF	0.6240*		0.4437		0.3133		0.2318	

Figure 8.5: The interpolated recall-precision in the ST instance



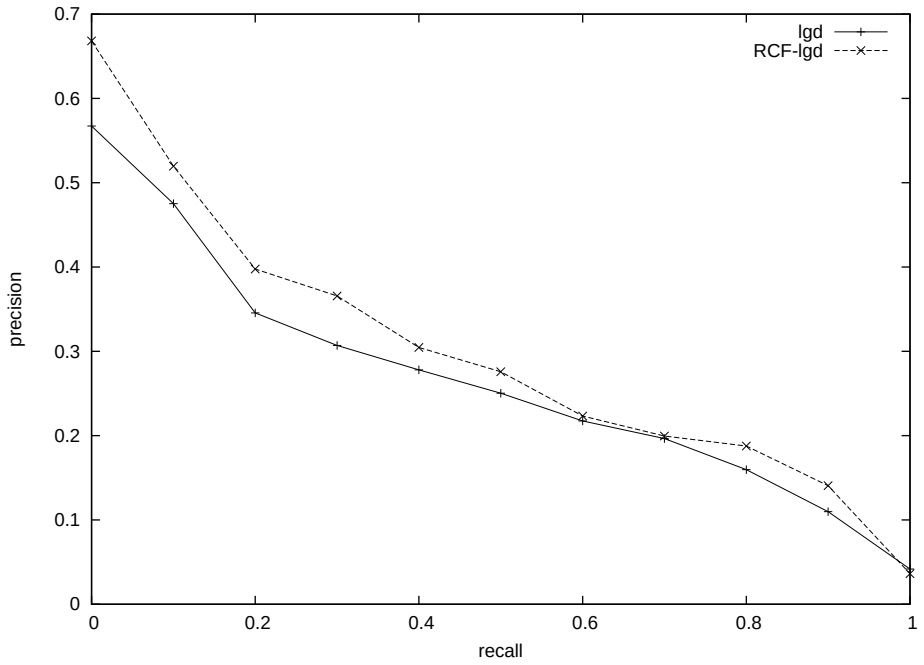
(a) $clef09$ (b) $clef10$

(c) $clef11$ (d) $clef12$

8.5.1 Discussion

We think that the ability of the RCF method to considerably improve the performance, can more and more increase by using more accurate mapping tools. Figure 8.7 shows the total number of phrases in the $clef10$ and $clef11$ corpora, and also shows the number of phrases with respect to the depth of the hierarchy that we can build for each phrase. More precisely, (Figure 8.7) shows that in about 86% of cases, the depth of phrases' hierarchies is 1, or in other words, in about 86% of cases we can not build the hierarchy of a phrase because this phrase is only one solid part and does not have sub-phrases, which means, the phrase is only one word. For example, in $clef10$ corpus, for 1916593 out of 2458245 phrases, we can not build a hierarchy of depth more than 1. Hence, it is clear that MetaMap, which is the mapping tool used in these experiments, maps only phrases of one word to UMLS' concepts. In other words, in 86% of cases MetaMap is not capable of recognizing the correct phrases in the text, and consequently it only makes a simple word-by-word text scan, and then it tries to map each word to UMLS' concepts. Without hierarchy or with a hierarchy of depth equals 1, our counting approach loses some of its effectiveness and gives a fix importance to each concept in the set of candidate concepts of a particular phrase. For example, assume the phrase (or word) '*xray*' only (Table

Figure 8.6: The interpolated recall-precision of lgd in $clef10$ using and without using RCF



6.1–P.113), each concept of its six candidate concepts will be assigned $\frac{1}{6}$ as a relative frequency, because there is no hierarchy.

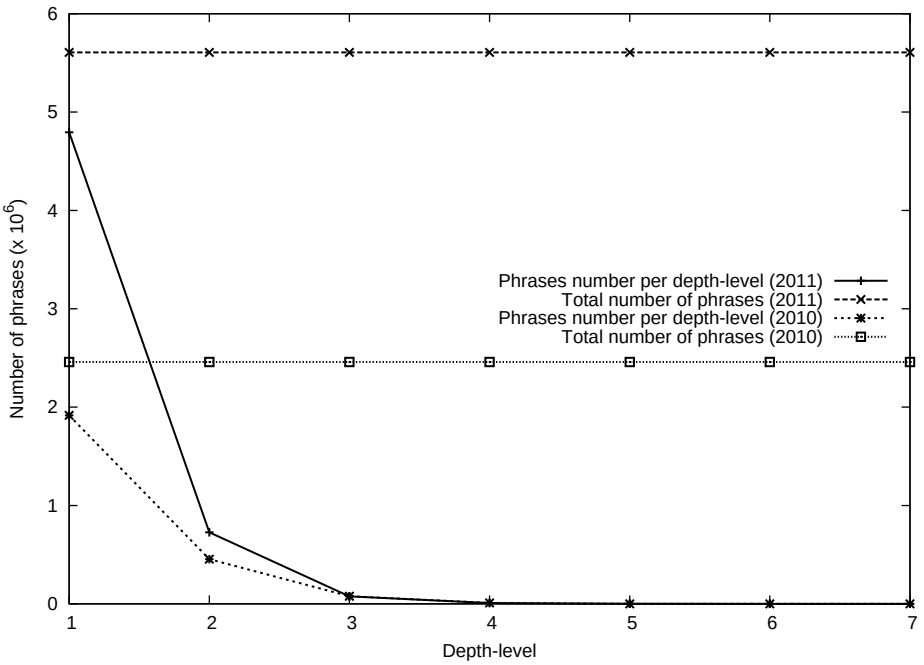
8.6 Conclusion

We presented in this chapter the experimental results of three instances ES, RL, and ST of our theoretical model. We have three groups of results, one for each instance. On important result of our experiments is to show that our logical IR model is efficiently applicable to large-scale data. This conclusion responds to the goal of this thesis that is declared in the general introduction (Chapter 1–P.3). To our knowledge, this is the first time that a logical model is applied to and tested on such large-scale corpora. We think that it is a non-marginal step in the field of logical IR models.

Additionally, the main purpose of comparing the experimental behavior of our logical model (with its current technical configurations) to the behavior of some high-performance IR models, is to place our model with respect to other models, and to draw some valuable conclusions about the way of making our model not only operational but also effective.

The experimental results of the ES instance, which simply integrates the new definitions of Exhaustivity and Specificity into one IR model, show that ES performs better than the high-performance baseline models that we choose. The outperformance of the ES instance is clearer when using concepts as indexing terms. In general, this group of experiments shows that integrating Exhaustivity and Specificity in an IR model could lead to a considerable gain in performance, and there should be a type of balance between Exhaustivity and Specificity. However, the relative weight of Exhaustivity and Specificity is still corpus-dependent.

Figure 8.7: The number of phrases with respect to the depth of the hierarchy of each phrase



The main goal of testing the RL instance, which exploits semantic relations between concepts, is to show the flexibility and the ability of our model to formally and efficiently integrate large-scale knowledge resources like UMLS into an IR model. This goal is effectively satisfied through the experiments that we managed on the RL instance. Additionally, this group of experiments shows that semantic relations aid in finding more relevant documents, but at the same time generate some noise. Therefore, it is hard to decide about using relations, and about the adequate semantic similarity measure that should be used. These conclusions are consistent with our theoretical expectation about document expansion (Equations 6.3&6.4–P.110). However, from our point of view, the experimental results of this instance are promising, because exploiting semantic relations between concepts (RL instance) performs better that the state-of-the-art models that consider the term-independence assumption, which is one of the main motivations of this thesis (Section 1.2.1–P.5).

Experiments on the ST instance, which excludes the term-independence assumption and exploits some inter-concepts hierarchical relations, show that the new term frequency approach, namely RCF, that respects the internal structure of phrases within text, is interesting and promising. Actually, the RCF approach of term-frequency computing showed an encouraging behavior when it is integrated into classical IR models. However, RCF is highly dependent on the quality of the output of mapping tools.

We should also clarify that, in general, the experimental results of any concept based IR model are highly depended on:

- The quality and the completeness of knowledge resources that contain concepts (UMLS in our case).
- The accuracy of text-concepts mapping tools (MetaMap in our case).

- The amount of information that is used besides concepts, such as relations between concepts. In other words, to which degree we profit from the content of knowledge resources that contain concepts.

Generally and based on the experimental results of the three instances of our model, we can say that our logical model is applicable to large-scale data, and is capable of integrating large knowledge resources. Our model also performs better than the baselines that we chose in this thesis. Furthermore, our model has potentials to perform even better, because the experimental results that we obtained in this chapter correspond to some basic configurations. Actually, there still exists some work to do especially when dealing with knowledge resources in IR. There is also the concept weighting issue, where we think that the classical term weighting mechanism is not the ideal one for weighting the concepts [Abdulahhad *et al.*, 2012b, 2013b; Bendersky *et al.*, 2010, 2011].

We think that the main obstacle in front of our model to perform better can be summarized as follows: Our model is originally based on a formal logic, and hence documents and queries are represented as logical sentences. Therefore, the essential part to build high-performance model is, from our point of view, finding an effective and efficient way to transform the textual content of documents and queries into logical sentences. In this chapter, we presented some simple ways to build logical sentences from text, but we still think that we need more effective ways, especially for the automatic identification of some logical connectives (e.g. negation $\neg$, disjunction $\vee$). Actually, this is the main reason for which we said that we *partially* solved the first problem of logical models (Section 1.3.1–P.8).

Part V

CONCLUSION & PERSPECTIVES

Chapter 9

Conclusions and Perspectives

9.1 Conclusions

The work presented in this thesis lies in the range of logic-based Information Retrieval (IR) models. These models are based on a logical framework to represent documents d, queries q, and to express the relevance from the system's point of view, where d and q are logical sentences, and the retrieval decision is a logical implication $d \rightarrow q$. Moreover, there is a need to an uncertainty measure $U(d \rightarrow q)$ to reflex the uncertainty located in every detail of IR.

There are two main motivations beyond the work in this thesis (Motivations 1&2 in Chapter 1–P.3). The first motivation is related to the well-accepted *term-independence* assumption, which is, from our point of view, a superficial and inadequate hypothesis because terms are normally connected to each others via some relations, e.g. synonymy, antonymy (opposition), hyponymy-hypernymy (specific-general), meronymy-holonymy (part-whole), etc. Term-independence assumption leads to the well-known *term-mismatch* problem. The second motivation is related to the *inferential* nature of the retrieval decision, where a document d indexed by a term t_1 is not directly relevant, from the system's point of view, to a query q indexed by another term t_2. However, if we know that t_1 and t_2 are synonymous, then based on d and this knowledge, q can be inferred. Hence, the simple term-based intersection between a document and a query is clearly insufficient mechanism for building effective IR models. Therefore, logic-based IR is a candidate track of work, where using formal logics in IR is twofold, on the one hand, formal logics are well adapted for knowledge representation, and then for building IR models being capable of formally integrating knowledge resources into the retrieval process. On the other hand, formal logics are powerful tools for simulating and modeling the inferential nature of the retrieval process.

Rather than dealing with the problems that could occur when using concepts and knowledge resources, even we have some publications in this context, actually, the main focus of this thesis is to study the shortcomings of current logical IR models (Problems 1&2&3 in Chapter 1–P.3), and to propose a logic-based IR model being capable of overcoming some of these problems and shortcomings.

Logic-based IR models normally propose complex and hard to obtain document and query representations. This is clearly the case in models that are based on more expressive logics than Propositional Logic ($\mathcal{PL}$), e.g. conceptual graph, possible world, etc. However, even in

$\mathcal{PL}$ based IR models there is such a type of problems, where $\mathcal{PL}$ based models either: 1- deal with the full spectrum of logical sentences, but with using a complex inference mechanism for retrieval [Crestani & Rijsbergen, 1995; Picard & Savoy, 2000]. Consequently, these models are inefficient with respect to execution time, where the inference mechanisms are complex algorithms, or 2- deal with a restricted category of logical sentences in order to obtain models having an acceptable execution time with respect to inference [Losada & Barreiro, 2001].

Concerning the retrieval decision $d \rightarrow q$, the problem is that: when using more expressive logics than $\mathcal{PL}$, the matching between d and q becomes hard to compute. For example, conceptual graph projection, logical imaging (specially when there is a big number of possible worlds), concepts subsumption, etc. Even using $\mathcal{PL}$, without restricting the logical sentences that could model d and q, can lead to a hard to compute matching. Furthermore, the uncertainty measure U is either ad-hoc, e.g. the distance between two possible worlds, the cost of changing one conceptual graph to another, etc., or it is hard to implement, e.g. probability distributions in logical imaging, positioning, etc.

To sum up, current logical IR models suffer from shortcomings at each level, where they propose hard to obtain document and query representations. Moreover, logical models generally either do not present an explicit definition of the IR implication $d \rightarrow q$, or present a non-operational definition, where it is almost impossible to be applied to large-scale data using the current technology. Concerning the way of measuring the uncertainty of the implication $U(d \rightarrow q)$, logical models sometimes present ad-hoc approaches, and in other times they present hard to implement measures. In this thesis, we address these shortcomings, and propose a new logical IR model that is to a large degree free of these shortcomings.

9.1.1 Theoretical Conclusions

In this thesis, we propose a logic-based IR model in order to overcome most of the previous limits. More precisely, we use $\mathcal{PL}$ as a mathematical framework. However, the new thing in our proposal, comparing to previous logic-based models, is that we exploit the potential mathematical link between formal logics and lattices in order to redefine the implication $d \rightarrow q$ and the uncertainty $U(d \rightarrow q)$.

At the level of document and query representation, our model represents d and q as logical sentences written in Disjunctive Normal Form (DNF), and *without any restriction*. Here, we remove the classical assumption, especially for documents, in logical IR models, which says, the logical sentence that represents the document d is the conjunction of the terms that appear in d (***partial solution of Problem 1***–P.8). The ability of our model to deal with any $\mathcal{PL}$ logical sentence without any restriction means that, we benefit from the full expressive power of $\mathcal{PL}$. However, we should not forget that the automatic identification of some logical connectives (e.g. negation $\neg$) from text faces some difficulties and still an open research question. Actually, although we present a simplified approach to automatically build some classes of logical sentences from text, the indexing process (identifying logical sentences from text) is beyond the scope of this thesis. Anyway, suppose that there is such an indexing process, our model is able to efficiently deal with any $\mathcal{PL}$ logical sentence, and that is important to build a logical model being capable of formally integrating knowledge resources into the IR process.

At the level of modeling the logical implication $d \rightarrow q$, first, we discuss that it is possible to use the material implication to represent the retrieval decision, and then we propose to replace

the implication $d \rightarrow q$ by the validity of material implication $\models d \supset q$. Showing that the material implication is an appropriate choice for IR is quite important, because using material implication means avoiding the other non-classical implications used in IR, which were one of the main obstacles to build operational logical IR models. Second, after proposing a new intermediate representation of $\mathcal{PL}$ logical sentences, we redefine the potential mathematical link between $\mathcal{PL}$ and lattices. According to this new link, $\mathcal{PL}$ logical sentences become nodes in a lattice of particular structure, and the partial order relation defined on this lattice becomes equivalent to the validity of material implication. This mapping between $\mathcal{PL}$ and lattices enables us to transform checking the validity of the material implication $d \supset q$ to a series of simple set-inclusion checking (***solution of Problem 2***–P.9). Checking the implication $d \rightarrow q$ was one of the main obstacles in front of logical models to become operational and applicable to large-scale data. Therefore, transforming the checking of $d \rightarrow q$ into a series of simple set-inclusion checking, through using the material implication and exploiting the mathematical link between formal logics and lattices, forms, from our point of view, an important step in the field of logical IR.

At the level of uncertainty $U(d \rightarrow q)$, since we position d and q on a lattice, and since we transform the material implication between two logical sentences to a partial order relation between their corresponding nodes in the lattice, then we suggest exploiting the *degree of inclusion or implication* function Z between two nodes of a lattice, which is introduced by Knuth [Knuth, 2005]. Using the degree of inclusion function Z to estimate the uncertainty of an implication allows us to define the uncertainty measure U as an intrinsic part of the logic (***solution of Problem 3***–P.9). Furthermore, the Z function has a mathematical basis, and thus the uncertainty measure U is no more ad-hoc. Additionally, there are many possible implementations of Z, and some of them are simple, e.g. the inner product of two vectors (Section 5.4.3.4–P.96). We think that estimating the uncertainty $U(d \rightarrow q)$ using a function having a mathematical basis, and at the same time having some simple implementations, forms also an important step in the field of logical IR. Moreover, the Z function has some interesting mathematical properties. For example, the function $Z(x, y)$ becomes equivalent to the conditional probability $P(x|y)$ if the function Z is consistent with all properties of distributive lattices. In fact, conditional probability plays an essential role in IR.

To sum up, we propose a logical IR model based on $\mathcal{PL}$ as a logical framework. We exploit the potential relation between $\mathcal{PL}$ and lattice theory, which allows us to, on the one hand, transform checking the validity of the logical implication $d \rightarrow q$ to a series of simple set-inclusion checking, on the other hand, exploit the degree of inclusion function Z defined on lattices to estimate the uncertainty $U(d \rightarrow q)$. Finally, our model is capable of working efficiently on any logical sentence without any restriction. It is also applicable to large-scale data, and that responds to the main goal of this thesis (Section 1.3–P.7). Our model also has some direct and appealing results, including:

- Formalizing and showing the adequacy of van Rijsbergen assumption about estimating the logical uncertainty $U(d \rightarrow q)$ through the conditional probability $P(q|d)$. Although this assumption was widely-accepted in the IR community, it is based on an intuition rather than a mathematical basis. Formalizing and providing a mathematical basis for this assumption was one of the direct results of our logical model, where we show the correctness of this assumption [Abdulahhad *et al.*, 2013a] (Section 5.4.1–P.89). To our

knowledge, this is the first study that formalizes the van Rijsbergen's assumption.

- Redefining Exhaustivity & Specificity. Using the degree of inclusion function Z to estimate the uncertainty U allows us to redefine Exhaustivity & Specificity in a way that the comparison between d and q is no more symmetric, where comparing d to q is different from comparing q to d. More precisely, to compute Exhaustivity we compare q with a part of it $d \wedge q$ (Equation 5.17–P.90), and to compute Specificity we compare d with a part of it $d \wedge q$ (Equation 5.18–P.91). On the one hand, to our knowledge, for the first time the two notions Exhaustivity & Specificity are comparable because they use the same object $d \wedge q$ to compare with (Equation 5.19–P.91). On the other hand, since we compare different objects to compute Exhaustivity and to compute Specificity, even using commutative functions (e.g. inner-product between two vectors) will give different values for Exhaustivity and for Specificity, and hence there is more flexibility to compute them.
- The possibility of reproducing most classical IR models as instances of our model. We showed in (Section 5.4.3–P.91) that our model is general enough to reproduce IR models of different mathematical bases. This general framework is useful for IR models comparison purposes. More importantly, our logical model, when seen as a general framework, shows the possibility to return almost all classical IR models to the same mathematical origin.

9.1.2 Experimental Conclusions

Theoretically, we showed that our model is able to overcome some problems of logical IR models. However, it is indispensable to support our theoretical solution by experimental evidence. Therefore, we also present, in this thesis, three operational instances inspired from our theoretical IR model. The first instance ES considers the flat document and query representation. This instance is applicable whatever the type of terms is, either words or concepts. In fact, ES shows the importance of integrating Exhaustivity & Specificity into an operational IR model. The second instance RL claims that terms are not independent. This instance is applicable to words or concepts, but, in this study, we restrict the instance to deal only with concepts. The instance exploits the isa relations between concepts to expand documents, in order to overcome the term-mismatch problem. In the context of this instance, we also theoretically show that document or query expansion generally improve recall but decrease precision. The third instance ST deals with the inadequacy of flat document and query representation, where it claims that there is a type of hierarchical relation between terms. Actually, this instance does not explicitly express this hierarchy, instead of that, it uses a new weighting system, namely Relative Concept Frequency (RCF), being capable of reflecting or modeling this hierarchical structure. This instance with its current configuration is only applicable to concepts.

Technically, we extract the list of words that index documents and queries after removing stop words and stemming using Porter algorithm. In addition, we use MetaMap to map the textual content of documents and queries to UMLS concepts, and to finally build the conceptual index of documents and queries. Since we use the vector space framework to implement our instances, we weight indexing terms, either words or concepts, using a variant of the classical *tf.idf* weighting schema. We also use a new concept counting approach, namely RCF, where

RCF takes the internal hierarchical structure of phrases into account. We apply our model to six corpora: four from ImageCLEF and two from TREC. We use the MAP and P@10 metrics for comparing the retrieval performance of our model with the performance of some high-performance baseline models. Actually, these baselines belong to a variety of mathematical frameworks, and that gives more credibility to the comparison. To check if one model is statistically better than another one, we use the Fisher's Randomization test at the 0.05 level. We use a variety of corpora and baselines in order to obtain more credible and useful conclusions.

Our experiments aimed to show that our logical model is operational and applicable to large-scale data (in accordance with the announced thesis' goal (Section 1.3–P.7)). However, we still compare the performance of the instances of our model with some high-performance models, in order to place our model with respect to the current state-of-the-art IR models. Anyway, managing experiments on corpora of the size of TREC and ImageCLEF, and exploiting knowledge resources of the size of UMLS, is a sufficient proof that our logical model is operational and applicable to large-scale data.

Concerning each instance alone, we manage experiments on the ES instance to show the importance of integrating both Exhaustivity & Specificity in one IR model. Concerning the RL instance, the goal is to show the flexibility and the ability of our model to formally and efficiently integrate large knowledge resources into an IR model. Our main intention beyond testing the ST instance is to show the validity of hypotheses that govern our new structure-based concept counting approach, namely RCF. In other words, in each instance we try to show one merit of our theoretical model.

The experimental results of the ES instance show that ES performs better than the high-performance baseline models that we choose. The outperformance of the ES instance is clearer when using concepts as indexing terms. In general, this group of experiments shows that integrating Exhaustivity and Specificity in an IR model could lead to a considerable gain in performance, and there should be a type of balance between Exhaustivity & Specificity. However, the relative weight of Exhaustivity & Specificity is still corpus-dependent. This experimental behavior of Exhaustivity & Specificity re-highlights their importance in IR. Exhaustivity & Specificity were abandoned for long time because there were not operational definitions of them.

Concerning the RL instance that exploits semantic relations between concepts, our experiments show that RL is comparable to the ES instance, and it outperforms our chosen baselines. This group of experiments shows that semantic relations aid in finding more relevant documents, but at the same time generate some noise. Therefore, it is hard to decide about using relations, and about the adequate semantic similarity measure that should be used. These conclusions are consistent with our theoretical expectation about document expansion (Section 6.4.2–P.110). Experiments on the RL instance show that our model is able to efficiently integrate large knowledge resources into the IR process. However, there still much work to do in this field, e.g. semantic similarity measures, expansion-term selection, expansion-term weighting, etc. Actually, this is beyond the scope of our thesis.

Experiments on the ST instance show that the new term frequency approach, namely RCF, that preserves the internal structure of phrases within text, is interesting. Actually, the RCF approach of term-frequency computing shows an encouraging behavior when it is integrated into classical IR models. The most important point in the RCF approach is that it maintains document and query lengths and consequently the corpus length. This property of RCF makes

it applicable with any classical IR model. However, RCF is highly dependent on the quality of the output of mapping tools (MetaMap in our case), where the more accurate the tool is in identifying phrases from text, the more ability to improve the performance the RCF approach has. Anyway, even with non very accurate tools, the RCF is still able to improve the performance, but maybe not as expected.

9.2 Perspectives

The work presented in this thesis can be further developed in the future in several ways and at several levels. Either at the level of our experiments (in the short-term), or even at the level of the theoretical model (in the long-term). Our goal in the short-term is essentially to investigate more experimental choices in order to build a rigid experimental basis. For example, applying our model to other corpora, using other knowledge resources, integrating other weighting systems, etc. However, our goal in the long-term is to explore some theoretical extensions of our model. Before that, we discuss in the following the potential influence of our model on the logical IR field in general.

As we already explained, our model is capable of efficiently dealing with any logical sentence and it is not restricted to one simplified category of sentences. On the one hand, that means our model takes advantage of the full expressive power of $\mathcal{PL}$, where it is possible to represent certain relations as logical implications. Therefore, in case that there is a powerful indexing process to extract logical sentences from text, it is possible to explore, to our knowledge for the first time, all capabilities of a logical IR model. On the other hand, it is possible to express some IR aspects like multilingualism, multi-modality, or late-fusion. For example, assume that the set of atomic propositions is $\Omega = W \cup C$ where W is a set of words and C is a set of concepts, and assume that d_w, q_w and d_c, q_c are the word-based and the concept-based index of d and q, respectively. If we put $d = d_w \vee d_c$ and $q = q_w \vee q_c$ then d will correspond to two nodes in our lattice and the same for q. By this way we could formally express the late-fusion aspect in textual IR between the word-based and the concept-based representations. To sum up, the ability of our model to efficiently deal with any logical sentence opens the door to either focus on building powerful indexing processes to transform text to logical sentences, or to re-model some IR aspects like multilingualism through representing documents and queries as several (more than one) nodes in our lattice.

Chapter 4 was dedicated to study the logical implication $d \rightarrow q$. At the end of that chapter and after a detailed discussion, we were able to propose using the material implication $d \supset q$ for IR (Hypothesis 4.1–P.74). Using material implication for IR releases researchers in the logical IR field from searching non-classical and quite complex definitions of $d \rightarrow q$. Furthermore, based on (Hypothesis 4.1) and based on our lattice, we transform checking the implication $d \rightarrow q$ to a series of simple set-inclusion checking. This transformation opens doors to use new and simple uncertainty measures, rather than Z. In this case, uncertainty measures will estimate the degree to which one set of elements includes another set of elements. This could form a new research area to find more effective and efficient uncertainty measures. Actually, this type of research in the logical IR field was not available, because the implication itself was difficult to verify.

Using the degree of inclusion function Z to estimate the uncertainty $U(d \rightarrow q)$ forms one

of the main choices in this thesis. Z is a general function and can be implemented in several ways. In this thesis we choose to implement Z as an inner-product of two vectors, but there still exists many other possible implementations that need to be explored. More importantly, in case that Z is consistent with all structural properties of distributive lattices, then Z is equivalent to conditional probability. We know that conditional probability plays an essential role in IR. Actually, we used this property of Z to formalize van Rijsbergen's assumption about replacing $U(d \rightarrow q)$ by $P(q|d)$ [Abdulahhad *et al.*, 2013a]. We also exploited the mathematical properties of Z for presenting new definitions of Exhaustivity & Specificity. In the same manner, the mathematical properties of Z can be explored either to formalize other IR notions or to build new IR matching functions.

The new definitions of Exhaustivity & Specificity, presented in this thesis, offer more practical forms of Exhaustivity & Specificity. Our experiments also show the importance of Exhaustivity & Specificity to build effective IR systems. Therefore, one possible and candidate track of research is to study if classical IR models implicitly integrate Exhaustivity & Specificity or not, and if not, how can these two notions be integrated into IR models?

We showed that our model is capable of reproducing most classical IR models as instances. Therefore, our model can form a general framework to theoretically understand and compare IR models. In addition, our model could be used to show that all IR models share the same mathematical origin. More precisely, our model is based on formal logics ($\mathcal{PL}$), probability (Z), and geometry (the lattice and the inner-product). These three mathematical theories are the main theories used in IR. Accordingly, our model could be seen as an intermediate between IR models and quantum mechanics [van Rijsbergen, 2004].

Our model is based on lattices. We also know that the expanded document d_e and query q_e result from adding some additional terms to the original document d and query q. Accordingly, the following order relations hold between d, q and d_e, q_e, where $d \leq d_e$ and $q \leq q_e$. Therefore, our model can be used as a theoretical framework to study expansion in a formal manner. This helps us to build better understanding of expansion. It could also help us to conclude some important directives to build an effective expansion method.

Some future research directions that are based on our logic and lattice based IR model presented in this thesis, include:

- Building an effective indexing process to automatically extract logical sentences from text, where our model is capable of dealing efficiently with any logical sentences, which is not the case in most logic-based IR models.
- Thinking about new and simple uncertainty measures to estimate $U(d \rightarrow q)$, where the list of available measures is now much bigger, because our model succeed to transform checking $d \rightarrow q$ into a series of simple set-inclusion checking.
- Studying the mathematical properties of Z and its influences on IR models. Studying also the different possible implementations of Z, where we used in this thesis one possible implementation (inner-product of two vectors). We also explored the mathematical properties of Z to formalize the van Rijsbergen's assumption and to redefine Exhaustivity & Specificity.
- Integrating Exhaustivity & Specificity into classical IR models after studying if these models implicitly integrate them or not, where we presented in this thesis new practical

definitions of Exhaustivity & Specificity, and we experimentally showed the importance of them.

- Building a general IR framework to understand and compare IR models, where we showed in this study that our model can reproduce other IR models.
- Showing that the model presented in this thesis could form a mediator between classical models and quantum mechanics, where our model is based on formal logics, probability, and geometry, which are the main mathematical theories used in IR, and they form the basis of quantum mechanics.
- Building a general framework to understand and study expansion, where there is a partial order relation between the expanded documents d_e and queries q_e on the one hand, and the original documents d and queries q on the other hand, where $d \leq d_e$ and $q \leq q_e$.

After long abandonment of the logical IR field, we hope that this study forms a valuable contribution to this field, and we hope that it forms a solid basis to bring light back to logical IR. In the following two subsections, we present our plan, in the short and long terms, to develop this study.

9.2.1 In the Short-Term

Using relations. We showed that exploiting semantic relations between concepts does not exceed the flat representation, where in some corpora using relations improves the performance, whereas in other corpora relations destruct the performance. This side of our work needs much more research, especially concerning the semantic relations that should be used, e.g. should we use only the isa relation, the *part-of* relation, etc., or all? In addition, the semantic similarity measure is a crucial part of models that use relations. Here also, there are many questions need to be answered, e.g. what is the measure that should be used with each relation? should we use the same measure with all relations? or it is preferable to choose a measure to each relation, etc. [Fang & Zhai, 2006].

Concept weighting. The Relative Concept Frequency (RCF) method studies the destruction that could happen when indexing using concepts instead of words, and it exceeds the flat document and query representation. The RCF method is related to the TF part of concept weighting. However, the other part of concept weighting, namely IDF, still untouched. On the one hand, Bendersky et al. [Bendersky *et al.*, 2011] show that only IDF is not sufficient to correctly weight concepts. They use several techniques to compensate the aspects not covered in IDF [Bendersky *et al.*, 2010, 2011]. On the other hand, we do not exploit the supplementary information contained in knowledge resources about concepts. For example, the position of a concept in the global conceptual hierarchy of the knowledge resource, where the hypothesis here is that deeper concepts are more important because they normally represent more precise notions. For example, in (Figure 7.3–P.123) the depth of the concept of '*B-cell*' is larger than the depth of the concept of '*Lymphocyte*', and the depth of the concept of '*Lymphocyte*' is larger than the depth of the concept of '*mononuclear Leukocyte*', and so on. By exploiting the available information in the knowledge resource, we think it is possible to build more effective IR models.

Text-concepts mapping tools. We showed that the new concept frequency computing method, namely RCF, is promising and it improves the retrieval performance. However, the usefulness of this method depends on the accuracy of MetaMap, which is the text-concept mapping tool used here. We plan in the foreseeable future to use other mapping tools, and study the accuracy of each tool and how it affects the retrieval performance.

In general, the experimental results of any concept based IR model are highly depended on: 1- the quality and the completeness of knowledge resources that contain concepts (UMLS in our case), 2- the accuracy of text-concepts mapping tools (MetaMap in our case), and 3- the amount of information that is used besides concepts, such as relations between concepts, concept depth, etc. In other words, to which degree we profit from the content of knowledge resources that contain concepts.

9.2.2 In the Long-Term

First, we said that one of the most important properties of formal logics is their ability to formally represent knowledge, and their ability to formally integrate this knowledge into IR. Assume that knowledge is represented by a set of logical sentences Γ, and assume that the retrieval decision between a document d and a query q is represented by the following inference $d \vdash q$. One of used ways to formally integrate knowledge into IR is to add Γ to the premises of the IR inference as follows: $\{\Gamma, d\} \vdash q$. If we project this idea on our model then the retrieval decision $\models d \supset q$ becomes $\Gamma \models d \supset q$ or equivalently $\models (\Gamma \wedge d) \supset q$. Actually, in this thesis, we implicitly integrate knowledge when we build operational instances of our model, e.g. indexing by concepts, exploiting relations between concepts, etc. Therefore, we should in the future study how knowledge Γ can be formally and explicitly integrated into our model. In other words, studing the implication $\Gamma \models d \supset q$ instead of the current implication $\models d \supset q$. Furthermore, studying the effects of this extension on the uncertainty measure Z.

Second, in order to estimate the uncertainty $U(d \rightarrow q)$, we use the degree of inclusion function Z. This function quantifies partial order relations defined on lattices. If Z is defined on a distributive lattice and it is consistent with all properties of distributive lattices then it is equivalent to a conditional probability, where $Z(x, y) = P(x|y)$. In addition, Z satisfies several rules (Appendix B–P.175), and it is possible to deduce some other rules. Therefore, we could, in future, work on these rules for more in-depth study of the theoretical properties of our model, or maybe finding new definitions of some theoretical IR notions like what we did with Exhaustivity and Specificity.

Third, we showed that most classical IR models are instances of our model, or in other words, our model is capable of reproducing most classical IR models. Hence, based on the properties of our model, we think it is possible to define some useful extensions of these classical models. For example, defining a variant of language models that take several views of documents and queries into account, where it is always possible to build different representations of a document or a query.

Fourth, in this thesis, we use the vector space mathematical framework in order to build a concrete and testable instances of our theoretical model. On the one hand, although interesting there are other noticeable and more modern mathematical frameworks than vector space are used in IR, e.g. language models, probabilistic models, etc. On the other hand, if Z is consistent with all properties of distributive lattices then it corresponds to a conditional probability.

Therefore, one possible development of this study is to instantiate our theoretical model using another mathematical framework like probability.

Finally, on the one hand, we studied in this thesis the two notions Exhaustivity and Specificity, and we propose a new mathematical definition of them. The experimental results of the ES instance of our model show that integrating both Exhaustivity and Specificity into a concrete IR model improves its performance. On the other hand, Fang et al. [Fang *et al.*, 2004] and Clinchant et al. [Clinchant & Gaussier, 2010, 2013] present an axiomatic analysis of classical IR models and pseudo-relevance feedback models. Therefore, we think that presenting a similar axiomatic study to analyze Exhaustivity and Specificity, in view of their new definitions in this thesis, will be valuable and useful. Actually, this axiomatic analysis should be able to show if an IR model correctly integrates these two notions or not, in order to finally extend it in an appropriate and useful way.

Chapter 10

Publications

10.1 International Peer-Reviewed Conferences

ABDULAHHAD, K., CHEVALLET, J.P. & BERRUT, C. (2013). Revisiting Exhaustivity and Specificity Using Propositional Logic and Lattice Theory. In *Proceedings of the 2013 Conference on the Theory of Information Retrieval*, ICTIR '13, 21:93–21:100, ACM, New York, NY, USA.

ABDULAHHAD, K., CHEVALLET, J.P. & BERRUT, C. (2013). Revisiting the Term Frequency in Concept-Based IR Models. In H. Decker, L. Lhotská, S. Link, J. Basl & A. Tjoa, eds., *Database and Expert Systems Applications*, vol. 8055 of *Lecture Notes in Computer Science*, 63–77, Springer Berlin Heidelberg.

ABDULAHHAD, K., CHEVALLET, J.P. & BERRUT, C. (2013). Is Uncertain Logical-Matching Equivalent to Conditional Probability? In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '13, 825–828, ACM, New York, NY, USA.

ABDULAHHAD, K., CHEVALLET, J.P. & BERRUT, C. (2012). The Effective Relevance Link between a Document and a Query. In S. Liddle, K.D. Schewe, A. Tjoa & X. Zhou, eds., *Database and Expert Systems Applications*, vol. 7446 of *Lecture Notes in Computer Science*, 206–218, Springer Berlin Heidelberg.

10.2 National Peer-Reviewed Conferences

ABDULAHHAD, K., CHEVALLET, J.P. & BERRUT, C. (2012). Matching Fusion with Conceptual Indexing. In C.R. et Jean-Pierre Chevallet, ed., *Actes du 4e Atelier Recherche d'Information SEmantique (RISE)*, 34–45, Bordeaux, France, session Système de Recherche d'Information Sémantique.

ABDULAHHAD, K., CHEVALLET, J.P. & BERRUT, C. (2011). Solving Concept Mismatch through Bayesian Framework by Extending UMLS Meta-Thesaurus. In G. Pasi & P. Bellot, eds., *CORIA*, 311–326, Éditions Universitaires d'Avignon.

ABDULAHHAD, K., CHEVALLET, J.P. & BERRUT, C. (2011). Exploiting and Extending a Semantic Resource for Conceptual Indexing. In *Troisième Atelier Recherche d'Information*

SEmantique, RISE'11, 22–28, Avignon, France.

10.3 Others

ABDULAHHAD, K., CHEVALLET, J.P. & BERRUT, C. (2013). A New Lattice-Based Information Retrieval Theory. Research Report RR-LIG-038, LIG, Grenoble, France.

ABDULAHHAD, K., CHEVALLET, J.P. & BERRUT, C. (2012). MRIM at ImageCLEF2012. From Words to Concepts: A New Counting Approach. In *CLEF (Online Working Notes / Labs / Workshop)*.

ABDULAHHAD, K., CHEVALLET, J.P. & BERRUT, C. (2011). Multi-Facet Document Representation and Retrieval. In *CLEF (Notebook Papers / Labs / Workshop)*.

Part VI

APPENDICES

Appendix A

Relative Concept Frequency

A.1 Introduction

We have a contribution in the field of knowledge-base IR models. We mainly tackle the concept-mismatch problem and the term-frequency destruction problem that happens when moving from the word-space to the concept-space.

The concept-mismatch problem normally results from the inconsistency and incompleteness of knowledge resources. For example, the term '*Osteoporotic*' does not map to any concept in UMLS (version 2012AA), and the two related terms '*Dermatofibroma*' and '*Dermatofibrosarcoma*' correspond to two different concepts '*C0002991*' and '*C0392784*', respectively, and there is no relation linking these two concepts.

We tackle the concept-mismatch problem in two ways. In the first way [Abdulahhad *et al.*, 2011a,c], in order to compensate the incompleteness of knowledge resources, we propose, if two concepts are not related by any relation, to automatically define a new relation between them depending on their shared words. We then use a Bayesian Network based IR model to see the improvement that we could obtain due to the new defined relations. In the second way [Abdulahhad *et al.*, 2011b, 2012a], we use the data fusion approach to compensate the incompleteness of knowledge resources. We build different representations of documents and queries using different types of indexing terms, e.g. ngrams of characters, words, concepts. By this way, if there is a mismatch at the level of concepts then it could be compensated at the level of words, and if there is a mismatch at the level of words then it could be compensated at the level of ngrams. For example, in the query number 16 '*images of dermatofibroma*' of the ad-hoc image-based retrieval track of ImageCLEF2010[1] evaluation campaign, on the one hand, the word '*images*' is not an useful retrieving term, because 14 out of 16 queries contain this word. On the other hand, the corpus does not contain the word '*dermatofibroma*', and the corresponding concept of '*dermatofibroma*' does not also belongs to any document in the corpus. Therefore, the only way to match the query number 16 against the corpus is by using ngram of characters as indexing terms.

The term-frequency destruction problem happens when using concepts, instead of words, to index the content of documents and queries. This problem is one of the side-effects of the conceptual annotation or mapping process, where each term is mapped to a set of candidate

[1] www.imageclef.org/2010

concepts, and then document and query lengths change in a non-consistent way when moving from the word-space to the concept-space [Abdulahhad *et al.*, 2012b, 2013b].

Concerning both conceptual ambiguity and term-frequency destruction problems, instead of disambiguation, we apply a new concept counting approach [Abdulahhad *et al.*, 2012b, 2013b]. In other words, instead of mapping each term to only one disambiguated concept, we maintain all candidate concepts of a particular term, and we give to each concept a relative frequency or count compatible with two main hypotheses:

- Concepts that correspond to longer text should receive larger relative frequency.
- The relative frequency of a concept should be inversely proportional to the number of candidate concepts of the text that it belongs to. The bigger the set of candidate concepts is, the smaller relative frequency its concepts receive, because more candidate concepts means that the corresponding text is more ambiguous.

In the rest of this chapter, we talk about our new concept counting approach, namely *Relative Concept Frequency (RCF)*, which is used to correct the destruction that happens when moving from words to concepts.

A.2 Revisiting Term Frequency in Case of Concepts

Indexing documents and queries using concepts, instead of word-based indexing, is an alternative approach, and it supposes to give a more meaningful indexing. However, this way of indexing needs to revisit some hypotheses of classical IR. Therefore, we here propose a new concept counting approach, namely RCF, which counts concepts with respect to their corresponding text in the documents or queries.

Besides the contribution to solve both conceptual ambiguity and term-frequency destruction problems that happen when indexing by concepts instead of words, RCF exceeds the flat representation of documents and queries through exploiting some structural relations between concepts, and that is essential for the ST instance of our model.

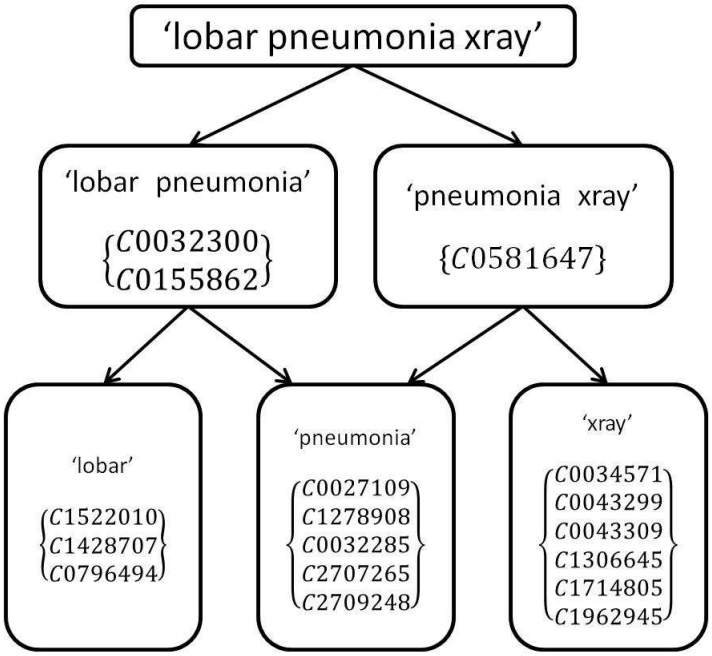
To obtain the concepts that correspond to a particular text, we need a mapping tool. In this study, we use MetaMap for mapping text to UMLS concepts. In general, mapping tools, not only MetaMap, first extract phrases from text, and then they map each phrase, or parts of it, to concepts. Finally, for each phrase, we get concepts corresponding to the phrase and its parts. Actually, this implicit structure between phrases and their parts is what we exploit to define our RCF approach that explicitly takes this structure into account. Table 6.1 (P.113) shows the UMLS concepts that correspond to the phrase '*lobar pneumonia xray*', or its parts, using MetaMap. The intuitive structure that can be built upon (Table 6.1) is depicted in (Figure A.1).

In classical bag of words based IR models, the weight of a word a in a document d is generally a consequence of Luhn conjecture [Luhn, 1958], and respects the two following rules:

- **Rule 1**: the weight of a is *proportional* to the frequency of a in d (descriptive measure).
- **Rule 2**: the weight of a is *inversely proportional* to the frequency of a in the corpus (discriminative measure).

Accordingly, our counting approach, namely RCF, respects the following hypotheses:

Figure A.1: The intuitive structure of the phrase '*lobar pneumonia xray*' using MetaMap



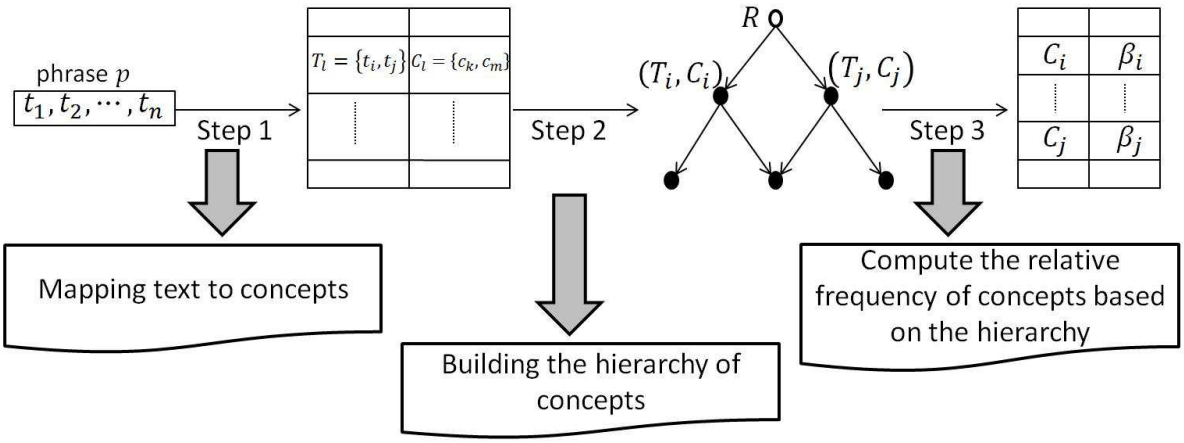
- **Hypothesis 1**: Concepts that correspond to longer text should receive larger relative frequency. For example, the relative frequency of '*C0032300*', which corresponds to the '*lobar pneumonia*' phrase, should be larger than '*C1522010*', which corresponds to the '*lobar*' phrase (Figure A.1).
- **Hypothesis 2**: The relative frequency of a concept c should be inversely proportional to the number of candidate concepts of the text that c belongs to. The bigger the set of candidate concepts is, the less important relative frequency its concepts receive, because more candidate concepts means that the corresponding text is more ambiguous. For example, each of the six concepts of '*xray*' (Figure A.1) should have less relative frequency than each of the two concepts of '*lobar pneumonia*'.
- **Hypothesis 3**: As in classical IR, the length of documents correspond to their textual length, and we here propose an approach that re-distributes the length of the document on its concepts (maintaining the document length).
- **Hypothesis 4**: Concepts of the same part of phrase are equally important. For example, the six concepts of '*xray*' are equally important.

These hypotheses validate the first rule of Luhn. In addition, our approach only affects the term frequency part. Therefore, concerning the second rule, it is easily conserved by using an IDF (Inverse Document Frequency) like measure.

A.2.1 Computing Relative Concept Frequency (RCF)

Mapping tools start by extracting phrases from text. Therefore, we explain our approach at phrase-level, and then we generalize it to document-level. For each phrase of document, map-

Figure A.2: The general process to compute RCF at phrase-level



ping tools generate its variants, or sub-phrases, and map these variants to sets of concepts (Step 1 Figure A.2). In fact, Step 1 represents the mapping tools. Based on the output of Step 1, we generate a phrase-hierarchy, which gives an overview of the concepts of the phrase (Step 2 Figure A.2). From the hierarchy of the concepts of a phrase, we compute the RCF of each concept (Step 3 Figure A.2). Globally, each phrase of each document is processed as described above. In the following, we present the definitions of the previous steps.

Step 1

Step 1 represents the work of mapping tools, where these mapping tools extract phrases, and for each phrase they search concepts that correspond to the phrase or parts of it. Since we use MetaMap in this study, then Step 1 represents MetaMap.

We define the set of words W, the set of phrases P, and the set of concepts C. Each phrase $p \in P$ is a sequence of words or equivalently a set of terms. We define the set of terms $T = W \times \mathbb{N}^*$, which is a set of tuples, and each tuple $(w, i) \in T$ links a word $w \in W$ with a number $i \in \mathbb{N}^*$. We also define the set of nodes $N = 2^T \times 2^C$, which links a set of terms with a set of concepts, where 2^T is the power set of T and 2^C is the power set of C.

We define two functions to link terms, phrases, and concepts. The function trm returns the set of terms that appear in a phrase, where

$$trm : P \rightarrow 2^T$$

For example, assume p is the '*lobar pneumonia xray*' phrase then:

$$trm(p) = \{(lobar, 1), (pneumonia, 2), (xray, 3)\}$$

where 1, 2, and 3 are the positions of the words in the phrase p, starting by 1. We define $|p|_T = |trm(p)|$ as the length of a phrase $p \in P$ in the word-space. For example, a phrase p like '*lobar pneumonia xray*' has a length $|p|_T = 3$.

The function map maps a phrase $p \in P$ to its parts and their concepts. This function fits to be a representation of any mapping tool, where

$$map : P \rightarrow 2^N$$

Table A.1: The output of applying the function map to the phrase '*lobar pneumonia xray*', where map stand for MetaMap

Terms	Candidate concepts
$T_1 = \{(lobar, 1), (pneumonia, 2)\}$	C_1=*{C0032300, C0155862}*
$T_2 = \{(pneumonia, 2), (xray, 3)\}$	C_2=*{C0581647}*
$T_3 = \{(lobar, 1)\}$	C_3=*{C1522010, C1428707, C0796494}*
$T_4 = \{(pneumonia, 2)\}$	C_4=*{C0024109, C1278908, C0032285 C2707265, C2709248}*
$T_5 = \{(xray, 3)\}$	C_5=*{C0034571, C0043299, C0043309 C1306645, C1714805, C1962945}*

$map(p)$ is the set of all parts of p with their candidate concepts. Therefore,

- $\forall p \in P, \forall (T_i, C_i) \in map(p), T_i \subseteq trm(p)$, or in other words, each part is a sub-phrase of the original phrase.
- $\forall p \in P, \forall (T_i, C_i) \in map(p)$, C_i are the set of concepts that the part T_i is mapped to.
- $\forall p \in P, \forall (T_i, C_i) \in map(p), C_i \neq \phi$, or in other words, we only consider the parts that have concepts.
- $\bigcap_{(T_i, C_i) \in map(p)} C_i = \phi$, the same concept does not appear more than one time in a phrase.

For example, assume p is the '*lobar pneumonia xray*' phrase then (see Table A.1):

$$map(p) = \{(T_1, C_1), (T_2, C_2), (T_3, C_3), (T_4, C_4), (T_5, C_5)\}$$

Step 2

We define a partial order relation $<$ on the set N, as follows:

$$\forall (T_i, C_i), (T_j, C_j) \in N, (T_j, C_j) < (T_i, C_i) \quad \textit{iff} \quad T_j \subset T_i$$

Using the partial order relation $<$, two functions ch and pr could be defined. However, we first define an abstract root node $R = (T_R, C_R)$, where $|T_R| = 0$, $|C_R| = 0$, and by definition $\forall (T_i, C_i) \in N, (T_i, C_i) < R$. The function ch returns the *direct children* of any node $n \in N \cup \{R\}$.

$$ch : N \cup \{R\} \rightarrow 2^N$$

where, $\forall (T_i, C_i), (T_j, C_j) \in N \cup \{R\}$ then $(T_j, C_j) \in ch((T_i, C_i))$ *iff*

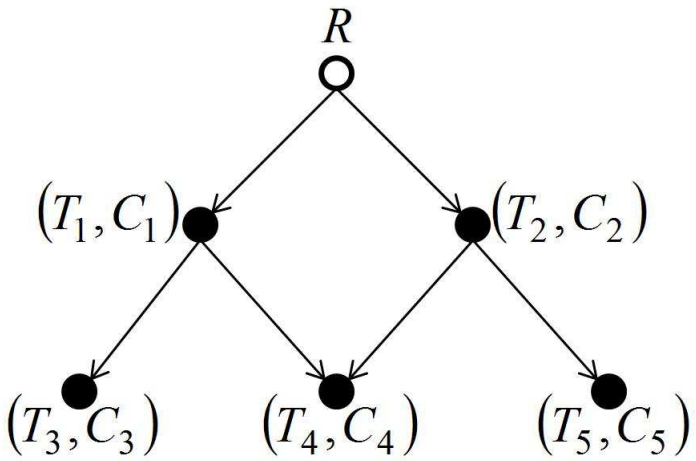
- $(T_j, C_j) < (T_i, C_i)$ and
- $\nexists (T_k, C_k) \in N \cup \{R\}$ satisfying that $(T_j, C_j) < (T_k, C_k) < (T_i, C_i)$

Reversely, we define the function pr that returns the *direct parents* of any node $n \in N \cup \{R\}$.

$$pr : N \cup \{R\} \rightarrow 2^{N \cup \{R\}}$$

where, $\forall (T_i, C_i), (T_j, C_j) \in N \cup \{R\}$ then $(T_j, C_j) \in pr((T_i, C_i))$ *iff*

Figure A.3: The hierarchy of the phrase '*lobar pneumonia xray*'



- $(T_i, C_i) < (T_j, C_j)$ and
- $\nexists (T_k, C_k) \in N \cup \{R\}$ satisfying that $(T_i, C_i) < (T_k, C_k) < (T_j, C_j)$

The hierarchy of a phrase $p \in P$ is defined by applying the two functions ch and pr to each node in $map(p) \cup \{R\}$. For example, assume p is the '*lobar pneumonia xray*' phrase. Figure A.3 shows the hierarchy that is defined on the set $map(p) \cup \{R\}$.

Step 3

We define the *Relative Frequency* function rf that relatively counts each concept of a phrase.

$$rf : P \rightarrow 2^{C \times \mathbb{R}^{+*}}$$

where $\forall p \in P, rf(p) = \{(c_1, \beta_1), \ldots, (c_r, \beta_r)\}$, c_i is a concept, β_i is the relative frequency of c_i. The function rf must respect the following points:

- $\bigcup_{(c_i,\beta_i) \in rf(p)} \{c_i\} = \bigcup_{(T_i,C_i) \in map(p)} C_i$, every concept of p must appear in $rf(p)$.
- $\forall (c_i, \beta_i) \in rf(p)$ and suppose that $(T_j, C_j) \in map(p)$ is the node that contains the concept c_i, where $c_i \in C_j$, then:
  - the relative frequency β_i of a concept c_i must be *proportional* to $|T_j|$ (Hypothesis 1).
  - the relative frequency β_i of a concept c_i must be *inversely proportional* to $|C_j|$ (Hypothesis 2).
- $\sum_{(c_i,\beta_i) \in rf(p)} \beta_i = |p|_T$, we maintain the length in both word-space and concept-space. Maintaining the length of phrases in both the word-space and concept-space implicitly leads to maintaining the length of document (Hypothesis 3).

The principle of computing rf is to build a hierarchy of the concepts of the phrase, and then the length of the phrase is distributed on the concepts respecting the four hypotheses, and the position of concepts within the hierarchy.

Assume a phrase $p \in P$ and the node $n = (T_n, C_n) \in map(p) \cup \{R\}$. The node n has $|pr(n)|$ parents and $|ch(n)|$ children. Each node n must distribute a certain amount α_n on its children. If n is the abstract root R then $\alpha_R = |p|_T$ the phrase length by default.

To compute rf, we attach three values $\alpha_i, \beta_i, \delta_i$ to each node $(T_i, C_i) \in map(p) \cup \{R\}$ in the hierarchy, where:

- α_i is the amount that must be distributed on the concepts of the current node $n_i = (T_i, C_i)$ and its children $ch(n_i)$. Since the relative frequency of a concept c in a node n_i must be proportional to $|T_i|$ (Hypothesis 1), then α_i must also be.

$$\alpha_i = \sum_{n_j \in pr(n_i)} \delta_j \times |T_i|$$

- δ_i is the portion of one single term of the input amount α_i.

$$\delta_i = \frac{\alpha_i}{|T_i| + \sum_{n_j \in ch(n_i)} |T_j|}$$

- β_i is the relative frequency of each concept $c \in C_i$. Remember that the relative frequency of a concept in a node n_i is proportional to $|T_i|$ (Hypothesis 1) and inversely proportional to $|C_i|$ (Hypothesis 2). Furthermore, concepts within the same node are equally important (Hypothesis 4).

$$\beta_i = \frac{\delta_i \times |T_i|}{|C_i|}$$

The algorithm starts from the abstract root R with $\alpha_R = |p|_T$. The algorithm must achieve a *breadth-first* search on the hierarchy that is defined on $map(p) \cup \{R\}$, and for each node the previous three values are computed in the following order: $\alpha_i \rightarrow \delta_i \rightarrow \beta_i$.

Finally, the length of a phrase p in the concept-space $|p|_C$ is equal to the sum of the relative frequencies of all concepts of p:

$$\forall p \in P, |p|_C = \sum_{(c_i, \beta_i) \in rf(p)} \beta_i$$

Knowing that the algorithm always starts at the abstract root and the input amount for the abstract root is the length of the phrase in the word-space; knowing that at any node, we do not produce any new amount, we just distribute the received amount on concepts and children; and knowing that at any node, we do not lose any amount, or in other words, the whole received amount is distributed on concepts and children, and for leaves the whole amount is distributed on concepts; then it is easy to verify that we maintain the length of phrases in both word-space and concept-space. In other words, we know that:

$$\forall p \in P, |p|_C = \sum_{(c_i, \beta_i) \in rf(p)} \beta_i = |trm(p)| = |p|_T$$

Returning to our previous example in (Figure A.4), the process starts in R with the input amount $\alpha_R = |p|_T = 3$. The scanning order of nodes will be $\langle R, n_1, n_2, n_3, n_4, n_5 \rangle$.

- At the abstract node R:
  - $\alpha_R = |p|_T = 3$ by default.
  - Compute δ_R, where $\delta_R = \frac{\alpha_R}{|T_R|+\sum_{n_j \in ch(R)} |T_j|} = \frac{\alpha_R}{|T_R|+|T_1|+|T_2|} = \frac{3}{4}$
  - β_R is not defined in R because R is an abstract node and does not contain any concept.
- At the node n_1:
  - Compute α_1, where $\alpha_1 = \sum_{n_j \in pr(n_1)} \delta_j \times |T_1| = \delta_R \times |T_1| = \frac{3}{2}$
  - Compute δ_1, where $\delta_1 = \frac{\alpha_1}{|T_1|+\sum_{n_j \in ch(n1)} |T_j|} = \frac{\alpha_1}{|T_1|+|T_3|+|T_4|} = \frac{3}{8}$
  - Compute β_1, where $\beta_1 = \frac{\delta_1 \times |T_1|}{|C_1|} = \frac{3}{8}$
- By continuing in this way, the final output of our algorithm will be:
  $rf(p) = \{(C0032300, \frac{3}{8}), (C0155862, \frac{3}{8}), (C0581647, \frac{3}{4}),$
  $(C1428707, \frac{1}{8}), (C1522101, \frac{1}{8}), (C0796494, \frac{1}{8}),$
  $(C0024109, \frac{3}{20}), (C1278908, \frac{3}{20}), (C2707265, \frac{3}{20}),$
  $(C2709248, \frac{3}{20}), (C0032285, \frac{3}{20}), (C0034571, \frac{1}{16}),$
  $(C0043299, \frac{1}{16}), (C0043309, \frac{1}{16}), (C1306645, \frac{1}{16}),$
  $(C1714805, \frac{1}{16}), (C1962945, \frac{1}{16})\}$
  We can verify that: $\sum_{(c_i,\beta_i) \in rf(p)} \beta_i = 3 = |ph|_T$

From the previous example, we can see that the concepts of less ambiguous and longest phrases have the *highest* relative frequency. Inversely, concepts of most ambiguous and shortest phrases have the *lowest* relative frequency.

Relative Concept Frequency at Document Level

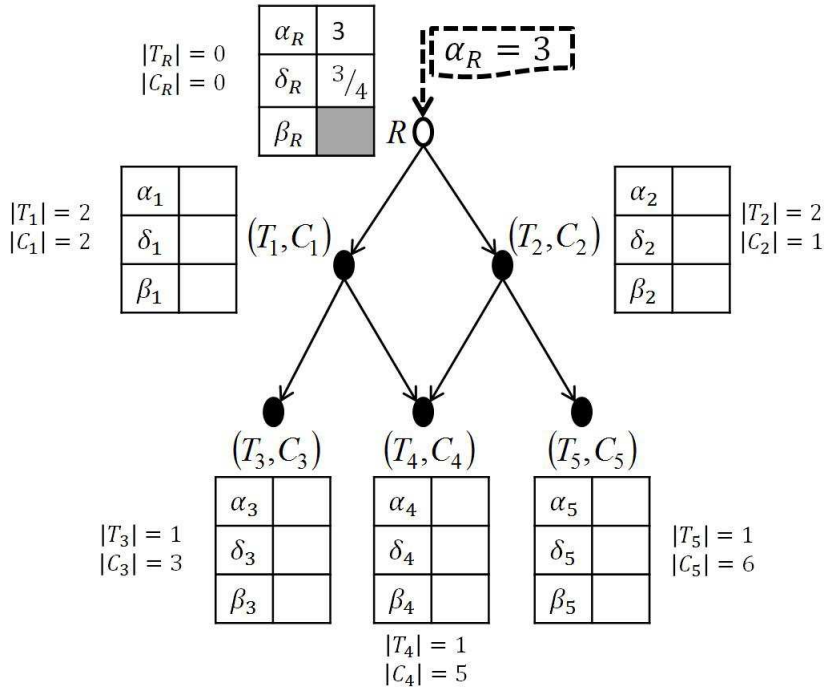
We presented how to compute the relative frequency of a concept at phrase-level. In this section, we show how it can be generalized at document-level. A document d is a sequence of phrases $P_d = \langle p_1, \ldots, p_{n_d} \rangle$. We represent the indexed document d as a set of concepts with their relative frequencies, where the relative frequency of a concept c in a document d, noted $rcf(c, d)$, is the sum of its relative frequencies within all phrases of d.

$$rcf(c, d) = \sum_{p_i \in P_d, (c,\beta_i) \in rf(p_i)} \beta_i \tag{A.1}$$

Queries are indexed as documents. In view of our document and query representation, the other weighting components become:

- The relative frequency of a concept c in a corpus D is the sum of the relative frequencies of c in all documents of D, $rcf(c, D) = \sum_{d_i \in D} rcf(c, d_i)$.
- Document length is $|d| = \sum_{c \in d} rcf(c, d)$. It is guaranteed that the document length is maintained $|d| = |trm(d)|$ (Hypothesis 3).
- Query length is $|q| = \sum_{c \in q} rcf(c, q)$. It is guaranteed that the query length is maintained $|q| = |trm(q)|$.
- Corpus length is $|D| = \sum_{d_i \in D} |d_i|$.

Figure A.4: The algorithm of computing rf function



A.3 Conclusion

We review in this appendix our contribution in knowledge-based IR. We briefly talk about our contribution to solve the concept-mismatch problem. We tackle this problem through compensating the incompleteness of knowledge resources by: 1- automatically adding some missing relations between concepts, and 2- using the data fusion approach.

We mainly talk, in this chapter, about the term-frequency destruction problem, which happens when using concepts instead of words, and how we tackle this problem. We propose a new concept counting approach, namely the Relative Concept Frequency (RCF). The central idea of the RCF approach is to re-distribute the length of a document on its concepts, and thus maintaining the document length in both the word-space and the concept-space. The distribution process is not random, but it respects some pre-defined hypotheses to finally validate the two famous weighting rules of Luhn.

The interest of the RCF approach is two-fold, where besides its contribution to solve the term-frequency destruction problem, it exceeds the flat representation of documents and queries through exploiting some structural relations between concepts, and that is essential for the ST instance of our model (Chapter 6–P.101). Mainly, RCF forms the basis of the weighting schema in the ST instance.

Appendix B

Lattice Theory

A lattice is an algebraic structure, or a set of elements satisfying certain properties. In the following, we present some definitions and examples related to lattices.

B.1 Definitions

Definition B.1 (Partially Ordered Set (poset))**.** *A partial order relation over a set of elements Ω is a binary relation $\leq_\Omega$, or simply $\leq$, satisfying the following conditions:*

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

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

Definition B.2 (Join $\veebar$ & Meet $\wedge$)**.** *Assume $(\Omega, \leq)$ is a poset. For any two elements $a, b \in \Omega$:*

1. *If a unique 'least upper bound' or 'the supremum' of a and b exists, it is called the* ***join****, denoted $a \veebar b$, and it satisfies the following conditions:*
   - $a \veebar b \in \Omega$
   - $a \leq (a \veebar b)$
   - $b \leq (a \veebar b)$
   - $\nexists c \in \Omega$, $c \neq (a \veebar b)$ *where* $a \leq c$ *and* $b \leq c$ *and* $c \leq (a \veebar b)$
2. *If a unique 'greatest lower bound' or 'the infimum' of a and b exists, it is called the* ***meet****, denoted $a \owedge b$, and it satisfies the following conditions:*
   - $a \owedge b \in \Omega$
   - $(a \owedge b) \leq a$
   - $(a \owedge b) \leq b$
   - $\nexists c \in \Omega$, $c \neq (a \owedge b)$ *where* $c \leq a$ *and* $c \leq b$ *and* $(a \owedge b) \leq c$

□

Definition B.3 (Lattice)**.** *A lattice* $(\Omega, \curlywedge, \curlyvee)$ *is defined either as a poset* $(\Omega, \leq)$ *where the join* $\curlyvee$ *and the meet* $\curlywedge$ *exist for each pair of elements in* Ω *or as an algebraic structure consisting of a set of elements* Ω *and two binary operations meet* $\curlywedge$ *and join* $\curlyvee$ *satisfying:*

1. *Idempotency:* $\forall a \in \Omega$, $a \curlywedge a = a$ *and* $a \curlyvee a = a$
2. *Commutativity:* $\forall a, b \in \Omega$, $a \curlywedge b = b \curlywedge a$ *and* $a \curlyvee b = b \curlyvee a$
3. *Associativity:* $\forall a, b, c \in \Omega$, $a \curlywedge (b \curlywedge c) = (a \curlywedge b) \curlywedge c$ *and* $a \curlyvee (b \curlyvee c) = (a \curlyvee b) \curlyvee c$
4. *Absorption:* $\forall a, b \in \Omega$, $a \curlywedge (a \curlyvee b) = a \curlyvee (a \curlywedge b) = a$

□

Definition B.4 (Distributive Lattice)**.** *Lattices that respect the following two conditions are called distributive lattices.*

1. *Distributivity of* $\curlywedge$ *over* $\curlyvee$*:* $\forall a, b, c \in \Omega$, $a \curlywedge (b \curlyvee c) = (a \curlywedge b) \curlyvee (a \curlywedge c)$
2. *Distributivity of* $\curlyvee$ *over* $\curlywedge$*:* $\forall a, b, c \in \Omega$, $a \curlyvee (b \curlywedge c) = (a \curlyvee b) \curlywedge (a \curlyvee c)$

□

Definition B.5 (Bounded Lattice)**.** *The algebraic structure* $(\Omega, \curlywedge, \curlyvee, \top, \bot)$ *is called a bounded lattice* iff $(\Omega, \curlywedge, \curlyvee)$ *is a lattice, and* $\top \in \Omega$ *and* $\bot \in \Omega$ *are the top and the bottom of* $(\Omega, \curlywedge, \curlyvee)$*, respectively, where:*

1. $\forall a \in \Omega$, $a \leq \top$ *and* $a \curlywedge \top = a$ *and* $a \curlyvee \top = \top$
2. $\forall a \in \Omega$, $\bot \leq a$ *and* $a \curlyvee \bot = a$ *and* $a \curlywedge \bot = \bot$

□

Definition B.6 (Complemented Lattice)**.** *If for any element* $a \in \Omega$ *in the bounded lattice* $(\Omega, \curlywedge, \curlyvee, \top, \bot)$*, there exists a unique element* $b \in \Omega$*, denoted* $b = \dot{\neg}\, a$*, satisfying:*

1. $a \curlywedge \dot{\neg}\, a = \bot$
2. $a \curlyvee \dot{\neg}\, a = \top$

then the algebraic structure $(\Omega, \curlywedge, \curlyvee, \dot{\neg}, \top, \bot)$ *is called a complemented lattice.* □

Definition B.7 (Boolean Algebra)**.** *Any distributive and complemented lattice* $(\Omega, \curlywedge, \curlyvee, \dot{\neg}, \top, \bot)$ *is a Boolean algebra.* □

Definition B.8 (Consistency Relations)**.** *In any lattice* $(\Omega, \curlywedge, \curlyvee)$*, the consistency relations explicitly express the relationship between the partial order relation* $\leq$ *and the meet* $\curlywedge$ *and the join* $\curlyvee$ *binary operations, as follows:*

$$\forall a, b \in \Omega, a \leq b \Leftrightarrow \begin{array}{l} a \curlywedge b = a \\ a \curlyvee b = b \end{array}$$

□

Definition B.9 (Sublattice)**.** *Assume that* $(\Omega, \curlywedge, \curlyvee)$ *is a lattice.* $(\Omega', \curlywedge, \curlyvee)$ *is a sublattice of* $(\Omega, \curlywedge, \curlyvee)$ iff*,*

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

□

Definition B.10 (Up-set)**.** *The up-set nodes of a node* x *in a lattice* $(\Omega, \wedge, \vee)$*, denoted* $\uparrow x$*, are:*

$$\uparrow x = \{x' | x' \in \Omega, x \leq x'\}$$

□

Definition B.11 (Down-set)**.** *The down-set nodes of a node* x *in a lattice* $(\Omega, \wedge, \vee)$*, denoted* $\downarrow x$*, are:*

$$\downarrow x = \{x' | x' \in \Omega, x' \leq x\}$$

□

B.2 The Degree of Inclusion

In any poset $(\Omega, \leq)$, the *zeta function* (Equation B.1) quantifies the notion of inclusion:

$$\forall a, b \in \Omega, \zeta(a, b) = \begin{cases} 1 & \text{if} \quad a \leq b \\ 0 & \text{if} \quad a \not\leq b \end{cases} \tag{B.1}$$

The function $\zeta(a, b)$ describes whether b includes a or not. Its dual function is:

$$\forall a, b \in \Omega, \zeta^{\partial}(a, b) = \begin{cases} 1 & \text{if} \quad a \geq b \\ 0 & \text{if} \quad a \not\geq b \end{cases} \tag{B.2}$$

The function $\zeta^{\partial}(a, b)$ describes whether a includes b or not.

For any two distinct elements a any b of a poset $(\Omega, \leq)$, if b includes a, $a \leq b$, then clearly a does not include b, $a \not\geq b$. However, even a does not include b, there is a way to describe the degree to which a includes b. Knuth [Knuth, 2005] generalized the inclusion (Equation B.2) to the degree of inclusion represented by real numbers. He introduced the Z *function*:

$$\forall a, b \in \Omega, Z(a, b) = \begin{cases} 1 & \text{if} \quad a \geq b \\ 0 & \text{if} \quad a \wedge b = \bot \\ z & \text{otherwise, where} \quad 0 < z < 1 \end{cases} \tag{B.3}$$

$Z(a, b)$ quantifies the degree to which a includes b. Knuth [Knuth, 2005] says: "The motivation here is that, if we are certain that a includes b then we want to indicate this knowledge. However, if we know that a does not include b, then we can quantify the *degree* to which a includes b".

Assume that instead of working with elements of poset $(\Omega, \leq)$, we work with elements of a *Distributive Lattice* $(\Omega, \wedge, \vee)$. In this case, if Z is consistent with the structure of distributive lattices then Z satisfies the following rules by which the degree of inclusion should be manipulated. For any distributive lattice $(\Omega, \wedge, \vee)$, $\forall a, b, c \in \Omega$:

1. *Sum* rule
$$Z(a \veebar b, c) = Z(a, c) + Z(b, c) - Z(a \barwedge b, c) \tag{B.4}$$
2. *First Product* rule
$$Z(a \barwedge b, c) = Z(a, c) + Z(b, c) - Z(a \veebar b, c) \tag{B.5}$$
3. *Second Product* rule
$$Z(a \barwedge b, c) = \alpha \times Z(a, c) \times Z(b, a \barwedge c) \tag{B.6}$$
where the constant α acts as a normalization factor.
4. *Bayes' Theorem* rule
$$Z(b, a \barwedge c) = \frac{Z(b, c) \times Z(a, b \barwedge c)}{Z(a, c)} \tag{B.7}$$

If the Z function is consistent with a Boolean algebra or a distributive and complemented lattice $(\Omega, \barwedge, \veebar, \barneg, \top, \bot)$ then Z corresponds to a conditional probability [Cox, 1946; Knuth, 2003, 2005]:
$$\forall a, b \in \Omega, Z(a, b) = P(a|b) \tag{B.8}$$
where P is a probability function.

B.2.1 Properties

Assume that Z is consistent with the structure of Boolean algebras then it satisfies: the *Sum* rule, the *First Product* rule, the *Second Product* rule, and the *Bayes' Theorem* rule.

B.2.1.1 The Chain Effect

Assume that there are three nodes x, y, t in a Boolean algebra where $x \leq y \leq t$. By applying the Bayes' Theorem rule and substituting a, b, c by y, x, t respectively, we get:
$$Z(x, t) = Z(x, y) \times Z(y, t) \tag{B.9}$$
If there is another element t' satisfying $x \leq y \leq t' \leq t$ then: $Z(x, t) = Z(x, y) \times Z(y, t') \times Z(y, t)$ and so on.

The *chain* effect means that for computing the degree of inclusion between any two comparable elements of a Boolean algebra, it is sufficient to compute the degree of inclusion of each connection on the order-path between theses two nodes.

B.2.1.2 The Multi-Path Effect

Assume the four nodes x, y, t_1, and t_2 of a Boolean algebra, where $t_1 \leq x \leq t_2$ and $t_1 \leq y \leq t_2$. Assume also that x and y are not comparable. There are two paths from t_1 to t_2. According to the chain effect (Equation B.9), we have: $Z(t_1, t_2) = Z(t_1, x) \times Z(x, t_2)$ and $Z(t_1, t_2) = Z(t_1, y) \times Z(y, t_2)$. Therefore, whatever is the path that you follow between any two nodes in a Boolean algebra, the degree of inclusion between these two nodes is the same. As a direct result of this effect, we have:
$$Z(x \barwedge y, x) \times Z(x, x \veebar y) = Z(x \barwedge y, y) \times Z(y, x \veebar y) \tag{B.10}$$

B.2.1.3 The Complement Effect

We know that, for any node x in a Boolean algebra, $x \veebar (\dot{\neg}\, x) = \top$. Therefore, for another node y, $Z(x \veebar (\dot{\neg}\, x), y) = 1$. Using the sum rule, we obtain:

$$Z(x, y) + Z(\dot{\neg}\, x, y) = 1 \tag{B.11}$$

B.3 Examples

Whatever Ω is a set of elements, it is possible to build several new Boolean algebras depending on Ω, as follows:

Theorem B.1. *The algebraic structure* $\mathcal{B}_1 = (2^\Omega \times 2^\Omega, \barwedge, \veebar, \dot{\neg}, \top, \bot)$ *is a Boolean algebra, where:*

1. 2^Ω *is the power set of* Ω
2. *meet operation:* $\forall (x_1, y_1), (x_2, y_2) \in 2^\Omega \times 2^\Omega, (x_1, y_1) \barwedge (x_2, y_2) = (x_1 \cap x_2, y_1 \cap y_2)$
3. *join operation:* $\forall (x_1, y_1), (x_2, y_2) \in 2^\Omega \times 2^\Omega, (x_1, y_1) \veebar (x_2, y_2) = (x_1 \cup x_2, y_1 \cup y_2)$
4. *complement operation:* $\forall (x, y) \in 2^\Omega \times 2^\Omega, \dot{\neg}\, (x, y) = (\Omega \setminus x, \Omega \setminus y)$
5. *top element:* $\top = (\Omega, \Omega)$
6. *bottom element:* $\bot = (\emptyset, \emptyset)$

The partial order relation $\leq$ *defined on* $\mathcal{B}_1$ *is:*

$$\begin{aligned} &\forall (x_1, y_1), (x_2, y_2) \in 2^\Omega \times 2^\Omega, \\ &\qquad [(x_1, y_1) \leq (x_2, y_2)] \Leftrightarrow [(x_1 \subseteq x_2) \textit{ and } (y_1 \subseteq y_2)] \end{aligned}$$

Proof. **Point 1.** The algebric structure $(2^\Omega \times 2^\Omega, \barwedge, \veebar)$ is a lattice, because (Definition B.3):

- Idempotency: $\forall (x, y) \in 2^\Omega \times 2^\Omega$
  $(x, y) \barwedge (x, y) = (x \cap x, y \cap y) = (x, y)$
  $(x, y) \veebar (x, y) = (x \cup x, y \cup y) = (x, y)$
- Commutativity: $\forall (x_1, y_1), (x_2, y_2) \in 2^\Omega \times 2^\Omega$
  $(x_1, y_1) \barwedge (x_2, y_2) = (x_1 \cap x_2, y_1 \cap y_2) = (x_2 \cap x_1, y_2 \cap y_1) = (x_2, y_2) \barwedge (x_1, y_1)$
  $(x_1, y_1) \veebar (x_2, y_2) = (x_1 \cup x_2, y_1 \cup y_2) = (x_2 \cup x_1, y_2 \cup y_1) = (x_2, y_2) \veebar (x_1, y_1)$
- Associativity: $\forall (x_1, y_1), (x_2, y_2), (x_3, y_3) \in 2^\Omega \times 2^\Omega$

$$\begin{aligned} (x_1, y_1) \barwedge [(x_2, y_2) \barwedge (x_3, y_3)] &= (x_1 \cap (x_2 \cap x_3), y_1 \cap (y_2 \cap y_3)) \\ &= ((x_1 \cap x_2) \cap x_3, (y_1 \cap y_2) \cap y_3) \\ &= [(x_1, y_1) \barwedge (x_2, y_2)] \barwedge (x_3, y_3) \\ (x_1, y_1) \veebar [(x_2, y_2) \veebar (x_3, y_3)] &= (x_1 \cup (x_2 \cup x_3), y_1 \cup (y_2 \cup y_3)) \\ &= ((x_1 \cup x_2) \cup x_3, (y_1 \cup y_2) \cup y_3) \\ &= [(x_1, y_1) \veebar (x_2, y_2)] \veebar (x_3, y_3) \end{aligned}$$

- Absorption: $\forall (x_1, y_1), (x_2, y_2) \in 2^\Omega \times 2^\Omega$

$$\begin{aligned}(x_1, y_1) \curlywedge [(x_1, y_1) \curlyvee (x_2, y_2)] &= (x_1 \cap (x_1 \cup x_2), y_1 \cap (y_1 \cup y_2)) \\ &= ((x_1 \cap x_1) \cup (x_1 \cap x_2), (y_1 \cap y_1) \cup (y_1 \cap y_2)) \\ &= (x_1 \cup (x_1 \cap x_2), y_1 \cup (y_1 \cap y_2)) \\ &= (x_1, y_1) \curlyvee [(x_1, y_1) \curlywedge (x_2, y_2)] \\ (x_1, y_1) \curlywedge [(x_1, y_1) \curlyvee (x_2, y_2)] &= (x_1 \cap (x_1 \cup x_2), y_1 \cap (y_1 \cup y_2)) \\ &= ((x_1 \cap x_1) \cup (x_1 \cap x_2), (y_1 \cap y_1) \cup (y_1 \cap y_2)) \\ &= (x_1 \cup (x_1 \cap x_2), y_1 \cup (y_1 \cap y_2)) \\ &= (x_1, y_1)\end{aligned}$$

Point 2. The lattice $(2^\Omega \times 2^\Omega, \curlywedge, \curlyvee)$ is distributive, because (Definition B.4):

- Distributivity of $\curlywedge$ over $\curlyvee$: $\forall (x_1, y_1), (x_2, y_2), (x_3, y_3) \in 2^\Omega \times 2^\Omega$

$$\begin{aligned}(x_1, y_1) \curlywedge [(x_2, y_2) \curlyvee (x_3, y_3)] &= (x_1 \cap (x_2 \cup x_3), y_1 \cap (y_2 \cup y_3)) \\ &= ((x_1 \cap x_2) \cup (x_1 \cap x_3), (y_1 \cap y_2) \cup (y_1 \cap y_3)) \\ &= [(x_1, y_1) \curlywedge (x_2, y_2)] \curlyvee [(x_1, y_1) \curlywedge (x_3, y_3)]\end{aligned}$$

- Distributivity of $\curlyvee$ over $\curlywedge$: $\forall (x_1, y_1), (x_2, y_2), (x_3, y_3) \in 2^\Omega \times 2^\Omega$

$$\begin{aligned}(x_1, y_1) \curlyvee [(x_2, y_2) \curlywedge (x_3, y_3)] &= (x_1 \cup (x_2 \cap x_3), y_1 \cup (y_2 \cap y_3)) \\ &= ((x_1 \cup x_2) \cap (x_1 \cup x_3), (y_1 \cup y_2) \cap (y_1 \cup y_3)) \\ &= [(x_1, y_1) \curlyvee (x_2, y_2)] \curlywedge [(x_1, y_1) \curlyvee (x_3, y_3)]\end{aligned}$$

Point 3. The lattice $(2^\Omega \times 2^\Omega, \curlywedge, \curlyvee)$ is bounded with the top element $\top = (\Omega, \Omega)$ and the bottom element $\bot = (\emptyset, \emptyset)$, because (Definition B.5):

- $\top \in 2^\Omega \times 2^\Omega$ and $\bot \in 2^\Omega \times 2^\Omega$
- We know that $\forall x, y \in 2^\Omega$ then $x \subseteq \Omega$ and $y \subseteq \Omega$ because 2^Ω is the power set of Ω. Therefore,
  $\forall (x, y) \in 2^\Omega \times 2^\Omega, (x, y) \leq \top$
  $\forall (x, y) \in 2^\Omega \times 2^\Omega, \bot \leq (x, y)$
- Top element: $\forall (x, y) \in 2^\Omega \times 2^\Omega$
  $(x, y) \curlywedge \top = (x, y) \curlywedge (\Omega, \Omega) = (x \cap \Omega, y \cap \Omega) = (x, y)$
  $(x, y) \curlyvee \top = (x, y) \curlyvee (\Omega, \Omega) = (x \cup \Omega, y \cup \Omega) = (\Omega, \Omega) = \top$
- Bottom element: $\forall (x, y) \in 2^\Omega \times 2^\Omega$
  $(x, y) \curlywedge \bot = (x, y) \curlywedge (\emptyset, \emptyset) = (x \cap \emptyset, y \cap \emptyset) = (\emptyset, \emptyset) = \bot$
  $(x, y) \curlyvee \bot = (x, y) \curlyvee (\emptyset, \emptyset) = (x \cup \emptyset, y \cup \emptyset) = (x, y)$

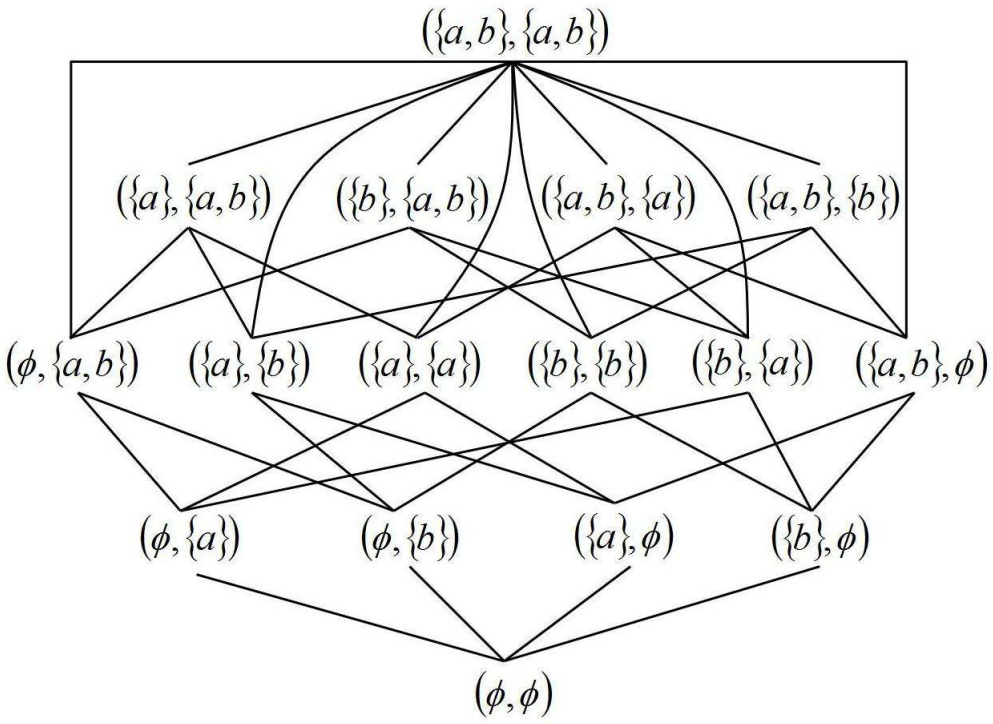
Point 4. The bounded lattice $(2^\Omega \times 2^\Omega, \curlywedge, \curlyvee, \top, \bot)$ is complemented, because (Definition B.6):

- We know that $\forall x \in 2^\Omega$ then $\Omega \setminus x \in 2^\Omega$ because 2^Ω is the power set of Ω. Therefore, $\forall (x, y) \in 2^\Omega \times 2^\Omega, (\Omega \setminus x, \Omega \setminus y) \in 2^\Omega \times 2^\Omega$
- $(x, y) \curlywedge \dot{\neg} (x, y) = (x, y) \curlywedge (\Omega \setminus x, \Omega \setminus y) = (x \cap (\Omega \setminus x), y \cap (\Omega \setminus y)) = (\emptyset, \emptyset) = \bot$
- $(x, y) \curlyvee \dot{\neg} (x, y) = (x, y) \curlyvee (\Omega \setminus x, \Omega \setminus y) = (x \cup (\Omega \setminus x), y \cup (\Omega \setminus y)) = (\Omega, \Omega) = \top$

According to (Definition B.7) and from points 1 & 2 & 3 & 4, we conclude that the algebraic structure $\mathcal{B}_1 = (2^\Omega \times 2^\Omega, \curlywedge, \curlyvee, \dot{\neg}, \top, \bot)$ is a Boolean algebra. □

Figure B.1 shows an example of the Boolean algebra $\mathcal{B}_1$ (Theorem B.1) when the set Ω contains two elements $\{a, b\}$.

Figure B.1: An example of the lattice $\mathcal{B}_1$ (Theorem B.1) when $\Omega = \{a, b\}$



Theorem B.2. *The algebraic structure* $\mathcal{B}_2 = (2^{2^\Omega}, \wedge, \vee, \dot{\neg}, \top, \bot)$ *is a Boolean algebra, where:*

1. 2^{2^Ω} *is the power set of the power set of* Ω.
2. *meet operation:* $\forall x, y \in 2^{2^\Omega}, x \wedge y = x \cap y$
3. *join operation:* $\forall x, y \in 2^{2^\Omega}, x \vee y = x \cup y$
4. *complement operation:* $\forall x \in 2^{2^\Omega}, \dot{\neg}\, x = 2^\Omega \setminus x$
5. *top element:* $\top = 2^\Omega$
6. *bottom element:* $\bot = \emptyset$

The partial order relation $\leq$ *defined on* $\mathcal{B}_2$ *is:*

$$\forall x, y \in 2^{2^\Omega}, [x \leq y] \Leftrightarrow [x \subseteq y]$$

Proof. **Point 1.** The algebric structure $(2^{2^\Omega}, \wedge, \vee)$ is a lattice, because (Definition B.3):

- Idempotency: $\forall x \in 2^{2^\Omega}$
  $x \wedge x = x \cap x = x$
  $x \vee x = x \cup x = x$
- Commutativity: $\forall x, y \in 2^{2^\Omega}$
  $x \wedge y = x \cap y = y \cap x = y \wedge x$
  $x \vee y = x \cup y = y \cup x = y \vee x$
- Associativity: $\forall x, y, z \in 2^{2^\Omega}$
  $x \wedge (y \wedge z) = x \cap (y \cap z) = (x \cap y) \cap z = (x \wedge y) \wedge z$
  $x \vee (y \vee z) = x \cup (y \cup z) = (x \cup y) \cup z = (x \vee y) \vee z$

- Absorption: $\forall x, y \in 2^{2^\Omega}$
  $x \barwedge (x \barvee y) = x \cap (x \cup y) = (x \cap x) \cup (x \cap y) = x \cup (x \cap y) = x \barvee (x \barwedge y)$
  $x \barwedge (x \barvee y) = x \cap (x \cup y) = (x \cap x) \cup (x \cap y) = x \cup (x \cap y) = x$

Point 2. The lattice $(2^{2^\Omega}, \barwedge, \barvee)$ is distributive, because (Definition B.4):

- Distributivity of $\barwedge$ over $\barvee$: $\forall x, y, z \in 2^{2^\Omega}$
  $x \barwedge (y \barvee z) = x \cap (y \cup z) = (x \cap y) \cup (x \cap z) = (x \barwedge y) \barvee (x \barwedge z)$
- Distributivity of $\barvee$ over $\barwedge$: $\forall x, y, z \in 2^{2^\Omega}$
  $x \barvee (y \barwedge z) = x \cup (y \cap z) = (x \cup y) \cap (x \cup z) = (x \barvee y) \barwedge (x \barvee z)$

Point 3. The lattice $(2^{2^\Omega}, \barwedge, \barvee)$ is bounded with the top element $\top = 2^\Omega$ and the bottom element $\bot = \emptyset$, because (Definition B.5):

- $\top \in 2^{2^\Omega}$ and $\bot \in 2^{2^\Omega}$
- We know that $\forall x \in 2^{2^\Omega}$ then $x \subseteq 2^\Omega$ because 2^{2^Ω} is the power set of 2^Ω. Therefore,
  $\forall x \in 2^{2^\Omega}, x \leq \top$
  $\forall x \in 2^{2^\Omega}, \bot \leq x$
- Top element: $\forall x \in 2^{2^\Omega}$
  $x \barwedge \top = x \barwedge 2^\Omega = x \cap 2^\Omega = x$
  $x \barvee \top = x \barvee 2^\Omega = x \cup 2^\Omega = 2^\Omega = \top$
- Bottom element: $\forall x \in 2^{2^\Omega}$
  $x \barwedge \bot = x \barwedge \emptyset = x \cap \emptyset = \emptyset = \bot$
  $x \barvee \bot = x \barvee \emptyset = x \cup \emptyset = x$

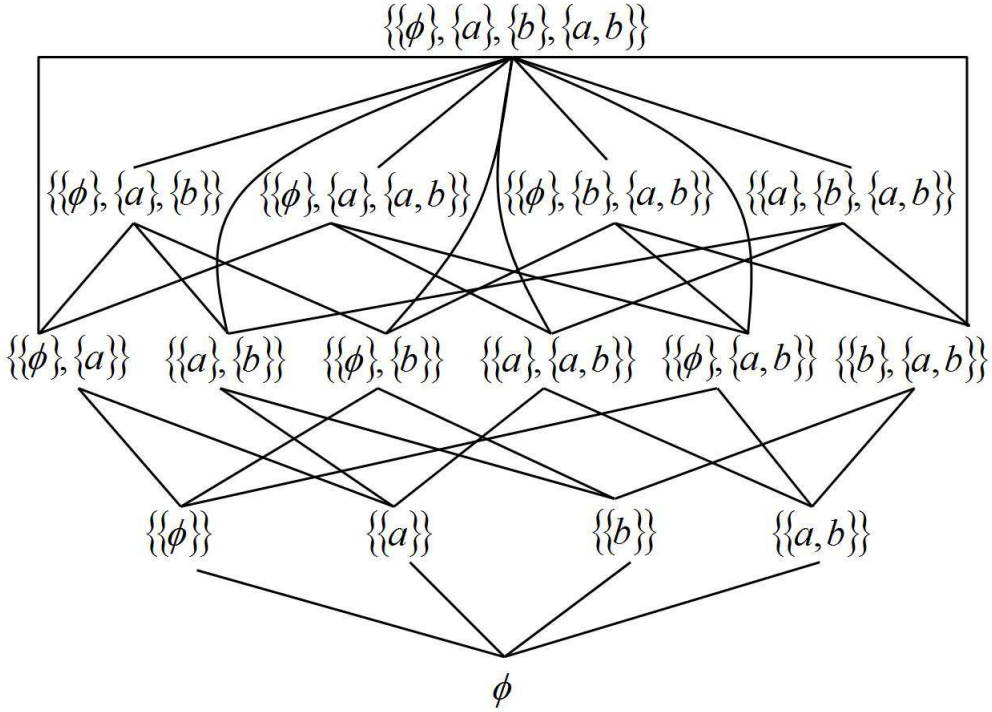
Point 4. The bounded lattice $(2^{2^\Omega}, \barwedge, \barvee, \top, \bot)$ is complemented, because (Definition B.6):

- We know that $\forall x \in 2^{2^\Omega}$ then $2^\Omega \setminus x \in 2^{2^\Omega}$ because 2^{2^Ω} is the power set of 2^Ω.
- $x \barwedge \dot{\neg} x = x \barwedge (2^\Omega \setminus x) = x \cap (2^\Omega \setminus x) = \emptyset = \bot$
- $x \barvee \dot{\neg} x = x \barvee (2^\Omega \setminus x) = x \cup (2^\Omega \setminus x) = 2^\Omega = \top$

According to (Definition B.7) and from points 1 & 2 & 3 & 4, we conclude that the algebraic structure $\mathcal{B}_2 = (2^{2^\Omega}, \barwedge, \barvee, \dot{\neg}, \top, \bot)$ is a Boolean algebra. □

Figure B.2 shows an example of the Boolean algebra $\mathcal{B}_2$ (Theorem B.2) when the set Ω contains two elements $\{a, b\}$.

Figure B.2: An example of the lattice $\mathcal{B}_2$ (Theorem B.2) when $\Omega = \{a, b\}$



Appendix C

Propositional Logic

C.1 Introduction

The propositional logic $\mathcal{PL}$, or zero-order logic, is defined on a finite set of atomic propositions or alphabet Ω and a finite set of connectives Υ. One of standard sets of connectives is $\Upsilon = \{\neg, \wedge, \vee\}$.

Definition C.1 (Alphabet Ω)**.** $\Omega = \{a_1, \ldots, a_n\}$ *is a finite set of atomic propositions, and it forms the alphabet of the logic.* □

Depending on the alphabet Ω and the connectives Υ, a set of well-formed logical sentences Σ is defined as follows:

- Any atomic proposition is a well-formed logical sentence: $\forall a \in \Omega, a \in \Sigma$.
- The negation of a logical sentence is also a logical sentence: $\forall s \in \Sigma, \neg s \in \Sigma$.
- The conjunction of any two logical sentences is also a logical sentence: $\forall s_1, s_2 \in \Sigma, s_1 \wedge s_2 \in \Sigma$.
- The disjunction of any two logical sentences is also a logical sentence: $\forall s_1, s_2 \in \Sigma, s_1 \vee s_2 \in \Sigma$.
- Σ does not contain any other sentences.

Material implication $\supset$ is implicitly included in Υ, because for any two logical sentences s_1 and s_2, the material implication $s_1 \supset s_2$ is equivalent to $\neg s_1 \vee s_2$.

Whatever the logic $\mathcal{L}$ is, not necessarily $\mathcal{PL}$, we say that a sentence s is *provable* based on a set of sentences Γ, denoted $\Gamma \vdash_{\mathcal{L}} s$ *iff* s can be obtained by applying the inference rules of $\mathcal{L}$ to the axioms of $\mathcal{L}$ and the set of sentences Γ. Moreover, $\vdash_{\mathcal{L}} s$ means that s can be obtained by applying the inference rules of $\mathcal{L}$ to only the axioms of $\mathcal{L}$.

$\mathcal{PL}$ defines a formal system consisting of a set of axioms and a set of inference rules, e.g. the Modus-Ponens rule: for any two sentences s_1 and s_2, if s_1 and $s_1 \supset s_2$ then s_2, or equivalently, $\{s_1, s_1 \supset s_2\} \vdash s_2$.

Table C.1: The set of interpretations based on $\Omega = \{a, b, c\}$

a	b	c	π	δ
F	F	F	$\pi_1 = \{(a, F), (b, F), (c, F)\}$	$\delta_1 = \{\}$
F	F	T	$\pi_2 = \{(a, F), (b, F), (c, T)\}$	$\delta_2 = \{c\}$
F	T	F	$\pi_3 = \{(a, F), (b, T), (c, F)\}$	$\delta_3 = \{b\}$
F	T	T	$\pi_4 = \{(a, F), (b, T), (c, T)\}$	$\delta_4 = \{b, c\}$
T	F	F	$\pi_5 = \{(a, T), (b, F), (c, F)\}$	$\delta_5 = \{a\}$
T	F	T	$\pi_6 = \{(a, T), (b, F), (c, T)\}$	$\delta_6 = \{a, c\}$
T	T	F	$\pi_7 = \{(a, T), (b, T), (c, F)\}$	$\delta_7 = \{a, b\}$
T	T	T	$\pi_8 = \{(a, T), (b, T), (c, T)\}$	$\delta_8 = \{a, b, c\}$

C.2 Formal Semantic

In $\mathcal{PL}$, a formal semantic or interpretation is given to a logical sentence s through assigning a truth value (T or F) to each atomic proposition in s. In other words, each interpretation corresponds to a mapping between all atomic propositions and the set of truth values $\{T, F\}$. The set of atomic propositions Ω thus corresponds to $2^{|\Omega|}$ possible interpretations because each atomic proposition $a \in \Omega$ is mapped to one of two possible values (T or F).

First, we define the set of all possible mappings Π_Ω between the set of atomic propositions Ω and the truth values $\{T, F\}$, where

$$\Pi_\Omega = \{\pi : \Omega \to \{T, F\} | \pi \text{ is a mapping}\}$$

where $|\Pi_\Omega| = 2^{|\Omega|}$.

Definition C.2 (Interpretations Δ_Ω)**.** *Each interpretation $\delta \in \Delta_\Omega$ is a subset of the atomic propositions Ω. It correponds to a mapping $\pi \in \Pi_\Omega$ between Ω and the truth values $\{T, F\}$. The set of interpretations Δ_Ω, which are based on a set of atomic propositions Ω, is:*

$$\Delta_\Omega = \{\delta = \{a \in \Omega | \pi(a) = T\} | \pi \textit{ is the corresponding mapping of } \delta\}$$

where $|\Delta_\Omega| = 2^{|\Omega|}$. □

The notation Δ_Ω is a simplification of Π_Ω, where: since atomic propositions can be mapped to only one of two possible truth values, then the mapping π can be simplified to the set of atomic propositions that are mapped to T, where the other atomic propositions are implicitly mapped to F.

Informally, for any alphabet Ω, the set of interpretations actually correspond to the different rows of the truth table that is built in terms of Ω. For example, suppose that Ω contains three propositions $\{a, b, c\}$, the truth table, the set of possible mappings Π_Ω, and the set of interpretations Δ_Ω are depicted in (Table C.1), where $\Pi_\Omega = \{\pi_1, \pi_2, \pi_3, \pi_4, \pi_5, \pi_6, \pi_7, \pi_8\}$ and $\Delta_\Omega = \{\delta_1, \delta_2, \delta_3, \delta_4, \delta_5, \delta_6, \delta_7, \delta_8\}$.

Any logical sentence $s \in \Sigma$ can be true of false in a specific interpretation $\delta \in \Delta_\Omega$. If s is true in δ, this is denoted $\{\delta\} \models s$ and is read as δ satisfies or models s, otherwise it is denoted

Table C.2: The truth table of the material implication $\supset$

a	b	δ	$a \supset b$	
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	$\{\delta_4\} \models a \supset b$

$\{\delta\} \not\models s$[1]. The truth value of an arbitrary logical sentence in Σ is determined with respect to an interpretation as follows:

- $\forall \delta \in \Delta_\Omega, \forall a \in \Omega, \{\delta\} \models a$ iff $a \in \delta$.
- $\forall \delta \in \Delta_\Omega, \forall s \in \Sigma, \{\delta\} \models \neg s$ iff $\{\delta\} \not\models s$.
- $\forall \delta \in \Delta_\Omega, \forall s_1, s_2 \in \Sigma, \{\delta\} \models s_1 \wedge s_2$ iff $\{\delta\} \models s_1$ and $\{\delta\} \models s_2$.
- $\forall \delta \in \Delta_\Omega, \forall s_1, s_2 \in \Sigma, \{\delta\} \models s_1 \vee s_2$ iff $\{\delta\} \models s_1$ or $\{\delta\} \models s_2$.

Definition C.3 (Models M)**.** *For any logical sentence s, the subset $M(s) \subseteq \Delta_\Omega$ that satisfy s is called the set of* models *of s, denoted $M(s) \models s$, where:*

$$M : \Sigma \rightarrow 2^{\Delta_\Omega}$$

For any interpretation $\delta \in M(s)$ we have $\{\delta\} \models s$, or equivalently, if we substitute each atomic proposition in s by its truth value in δ then the truth value of s will be true.

*The notation $\models s$ means that s is a tautology or **valid**, or in other words, s is true under any interpretation ($M(s) = \Delta_\Omega$). The notation $\not\models s$ means that s is false under all interpretations or **unsatisfiable** ($M(s) = \emptyset$).* □

For example, assume that $\Omega = \{a, b\}$ and assume that the logical sentence is the material implication $a \supset b$. The set of models $M(a \supset b) \models a \supset b$ is depicted in (Table C.2), where $M(a \supset b) = \{\delta_1, \delta_2, \delta_4\}$.

The set of models $M(s)$ is the set of models of a set of sentences logically equivalent to s. For example, $M(a \supset b) = M(\neg a \vee b)$ where it is well-known that $a \supset b$ and $\neg a \vee b$ are equivalent.

There are two main assumptions concerning the atomic propositions that do not occur in a sentence:

- Close World Assumption (CWA): for a sentence s, if an atomic proposition does not occur in s then it is implicitly false. For example, assume $\Omega = \{a, b, c\}$ then under CWA $M(a \wedge b) = \{\delta_7\}$ (Table C.1), where c must be false.
- Open World Assumption (OWA): for a sentence s, if an atomic proposition does not occur in s then it can be either true or false. For example, assume $\Omega = \{a, b, c\}$ then under OWA $M(a \wedge b) = \{\delta_7, \delta_8\}$ (Table C.1), where c can be true or false.

[1]The two symbols $\vdash$ and $\models$ are metalanguage symbols and they are not a part of the formal language of the logic.

$\mathcal{PL}$ is a special kind of formal logics where it is *complete* and *sound*. Completeness and soundness govern the relation between provability $\vdash$ and satisfiability $\models$. Completeness means that if $M(s_1) \models s_2$ then $s_1 \vdash s_2$. Soundness means that if $s_1 \vdash s_2$ then $M(s_1) \models s_2$.

Theorem C.1. *For any two logical sentences s_1 and s_2 we have:*

$$[\models s_1 \supset s_2] \Leftrightarrow [M(s_1) \subseteq M(s_2)]$$

Proof. 1. $s_1 \supset s_2$ is valid means that must not be there any model of s_1 which is not a model of s_2, or in other words, the validity of $s_1 \supset s_2$ means that in any interpretation if s_1 is true then s_2 must be also true, otherwise $s_1 \supset s_2$ is not valid.

$$[\models s_1 \supset s_2] \Rightarrow [M(s_1) \subseteq M(s_2)]$$

2. According to the definition of the material implication, $s_1 \supset s_2$ is true when either s_1 is false or both s_1 and s_2 are true. Therefore, if $M(s_1) \subseteq M(s_2)$ then $s_1 \supset s_2$ is valid or always true, because $M(s_1) \subseteq M(s_2)$ means that when s_1 is true then s_2 is also true.

$$[\models s_1 \supset s_2] \Leftarrow [M(s_1) \subseteq M(s_2)]$$

□

Theorem C.2 ($\mathcal{PL}$ vs. Lattices)**.** *The algebraic structure $\mathcal{B}_M = (2^{\Delta_\Omega}, \wedge, \vee, \dot{\neg}, \top, \bot)$ is a Boolean algebra, where:*

1. *2^{Δ_Ω} is the power set of Δ_Ω, and each element of 2^{Δ_Ω} is a possible set of interpretations. In other words, each element is a set of models of a set of logically equivalent sentences.*
2. ***meet** operation: $\forall x, y \in 2^{\Delta_\Omega}, x \wedge y = x \cap y$*
3. ***join** operation: $\forall x, y \in 2^{\Delta_\Omega}, x \vee y = x \cup y$*
4. ***complement** operation: $\forall x \in 2^{\Delta_\Omega}, \dot{\neg}\, x = \Delta_\Omega \setminus x$*
5. ***top** element of $\mathcal{B}_M$: $\top = \Delta_\Omega$.*
6. ***bottom** element of $\mathcal{B}_M$: $\bot = \emptyset$.*

The partial order relation $\leq$ defined on $\mathcal{B}_M$ is:

$$\forall x, y \in 2^{\Delta_\Omega}, [x \leq y] \Leftrightarrow [x \subseteq y]$$

Proof. The proof of this theorem can be directly established depending on (Theorem B.2). □

From (Theorems C.1 and C.2), the partial order relation on $\mathcal{B}_M$ is equivalent to the validity of material implication.

Theorem C.3. *For any three logical sentences s_1, s_2, s_3, we have:*

1. ***If** $s_1 \supset s_2$ is valid and $s_2 \supset s_3$ is valid **then** $s_1 \supset s_3$ is also valid.*
2. ***If** $s_1 \supset s_2$ is unsatisfiable and $s_2 \supset s_3$ is valid **then** $s_1 \supset s_3$ is not valid.*
3. ***If** $s_1 \wedge s_2$ is unsatisfiable **then** $s_1 \supset s_2$ is not valid.*

Proof. The proof of this theorem can be directly established depending on (Table C.3). □

Table C.3: Implications truth table

s_1	s_2	s_3	$s_1 \wedge s_2$	$s_1 \supset s_2$	$s_2 \supset s_3$	$s_1 \supset s_3$
F	F	F	F	T	T	T
F	F	T	F	T	T	T
F	T	F	F	T	F	T
F	T	T	F	T	T	T
T	F	F	F	F	T	F
T	F	T	F	F	T	T
T	T	F	T	T	F	F
T	T	T	T	T	T	T

Bibliography

ABDULAHHAD, K., CHEVALLET, J.P. & BERRUT, C. (2011a). Exploiting and Extending a Semantic Resource for Conceptual Indexing. In *Troisième Atelier Recherche d'Information SEmantique*, RISE'11, 22–28, Avignon, France. 7, 165

ABDULAHHAD, K., CHEVALLET, J.P. & BERRUT, C. (2011b). Multi-facet document representation and retrieval. In *CLEF (Notebook Papers/Labs/Workshop)*. 165

ABDULAHHAD, K., CHEVALLET, J.P. & BERRUT, C. (2011c). Solving concept mismatch through bayesian framework by extending umls meta-thesaurus. In G. Pasi & P. Bellot, eds., *CORIA*, 311–326, Éditions Universitaires d'Avignon. 7, 122, 165

ABDULAHHAD, K., CHEVALLET, J.P. & BERRUT, C. (2012a). Matching Fusion with Conceptual Indexing. In C.R. et Jean-Pierre Chevallet, ed., *Actes du 4e Atelier Recherche d'Information SEmantique (RISE)*, 34–45, Bordeaux, France, session Système de Recherche d'Information Sémantique. 165

ABDULAHHAD, K., CHEVALLET, J.P. & BERRUT, C. (2012b). Mrim at imageclef2012. from words to concepts: A new counting approach. In *CLEF (Online Working Notes/Labs/Workshop)*. 7, 114, 147, 166

ABDULAHHAD, K., CHEVALLET, J.P. & BERRUT, C. (2013a). Is uncertain logical-matching equivalent to conditional probability? In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '13, 825–828, ACM, New York, NY, USA. 71, 153, 157

ABDULAHHAD, K., CHEVALLET, J.P. & BERRUT, C. (2013b). Revisiting the term frequency in concept-based ir models. In H. Decker, L. Lhotská, S. Link, J. Basl & A. Tjoa, eds., *Database and Expert Systems Applications*, vol. 8055 of *Lecture Notes in Computer Science*, 63–77, Springer Berlin Heidelberg. 7, 114, 142, 147, 166

ALBITAR, S. (2013). *On the use of semantics in supervised text classification: application in the medical domain*. Ph.D. thesis, Ecole Doctorale en Mathématique et Informatique de Marseille. 30

AMATI, G. & OUNIS, I. (2000). Conceptual graphs and first order logic. *Comput. J.*, **43**, 1–12. 51, 55

AMATI, G. & VAN RIJSBERGEN, C.J. (2002). Probabilistic models of information retrieval based on measuring the divergence from randomness. *ACM Trans. Inf. Syst.*, **20**, 357–389. 130, 135, 138

AMATI, G., CASTIGLIONE, B. & KERPEDJIEV, S. (1992). AN INFORMATION RETRIEVAL LOGIC MODEL: IMPLEMENTATION AND EXPERIMENTS. *IEEE Transactions on Reliability*. 48

ARONSON, A.R. (2006). Metamap: Mapping text to the umls metathesaurus. *Bethesda MD NLM NIH DHHS*, 1–26. 23, 25, 28, 29, 120

ARONSON, A.R. & RINDFLESCH, T.C. (1997). Query expansion using the umls metathe-

saurus. *Proceedings of the AMIA Annual Fall Symposium*, 485–489. 24

ASLAM, J.A. & FROST, M. (2003). An information-theoretic measure for document similarity. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, SIGIR '03, 449–450, ACM, New York, NY, USA. 22, 122

BAADER, F. (2009). Description logics. In S. Tessaris, E. Franconi, T. Eiter, C. Gutierrez, S. Handschuh, M.C. Rousset & R. Schmidt, eds., *Reasoning Web. Semantic Technologies for Information Systems*, vol. 5689 of *Lecture Notes in Computer Science*, 1–39, Springer Berlin Heidelberg. 54

BAADER, F., CALVANESE, D., MCGUINNESS, D.L., NARDI, D. & PATEL-SCHNEIDER, P.F., eds. (2003). *The Description Logic Handbook: Theory, Implementation, and Applications*, Cambridge University Press. 6, 7, 17, 39

BARTELL, B.T., COTTRELL, G.W. & BELEW, R.K. (1994). Automatic combination of multiple ranked retrieval systems. In *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '94, 173–181, Springer-Verlag New York, Inc., New York, NY, USA. 24

BARWISE, J. (1989). *The Situation in Logic*. CSLI Lecture Notes, Center for the Study of Language and Information. 7, 39, 56

BARWISE, J. & PERRY, J. (1983). *Situations and Attitudes*. Bradford Books, Massachusetts Institute of technology. 7, 39, 56

BAZIZ, M. (2005). *Indexation conceptuelle guidée par ontologie pour la recherche d'information*. Thèse de doctorat, Université Paul Sabatier, Toulouse, France. 23, 24, 25, 28, 31, 37, 120

BELKIN, N.J., COOL, C., CROFT, W.B. & CALLAN, J.P. (1993). The effect multiple query representations on information retrieval system performance. In *Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '93, 339–346, ACM, New York, NY, USA. 24

BELKIN, N.J., KANTOR, P., FOX, E.A. & SHAW, J.A. (1995). Combining the evidence of multiple query representations for information retrieval. *Inf. Process. Manage.*, **31**, 431–448. 24

BĚLOHLÁVEK, R. & KLIR, G. (2011). *Concepts and fuzzy logic*. University Press Group Limited. 18, 19, 88, 103

BENDERSKY, M., METZLER, D. & CROFT, W.B. (2010). Learning concept importance using a weighted dependence model. In *Proceedings of the Third ACM International Conference on Web Search and Data Mining*, WSDM '10, 31–40, ACM, New York, NY, USA. 147, 158

BENDERSKY, M., METZLER, D. & CROFT, W.B. (2011). Parameterized concept weighting in verbose queries. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, SIGIR '11, 605–614, ACM, New York, NY, USA. 20, 21, 147, 158

BENDERSKY, M., METZLER, D. & CROFT, W.B. (2012). Effective query formulation with multiple information sources. In *Proceedings of the Fifth ACM International Conference on Web Search and Data Mining*, WSDM '12, 443–452, ACM, New York, NY, USA. 30

BERNERS-LEE, T., HENDLER, J. & LASSILA, O. (2001). The semantic web. *Scientific American*, **284**, 34–43. 54

BODENREIDER, O., BURGUN, A., BOTTI, G., FIESCHI, M., LE BEUX, P. & KOHLER, F. (1998). Evaluation of the unified medical language system as a medical knowledge source. *J Am Med Inform Assoc*, **5**, 76–87. 7, 22, 23, 37

BODENREIDER, O., BURGUN, A. & RINDFLEISCH, T. (2001). Lexically-suggested hyponymic relations among medical terms and their representation in the umls. In *Proceedings of Terminology and Artificial Intelligence, TIA'2001*, 11–21, Nancy, France. 7, 22, 23, 37

BOUGHANEM, M. & SOULÉ-DUPUY, C. (1992). A connexionist model for information retrieval. In A.M. Tjoa & I. Ramos, eds., *Database and Expert Systems Applications*, 260–265, Springer Vienna. 31

BRUZA, P.D. & VAN DER GAAG, L.C. (1993). Efficient context-sensitive plausible inference for information disclosure. In *Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '93, 12–21, ACM, New York, NY, USA. 113

BUCKLEY, C., SALTON, G., ALLAN, J. & SINGHAL, A. (1994). Automatic Query Expansion Using SMART: TREC 3. In *TREC*, 0+. 21

CARPINETO, C. & ROMANO, G. (2005). Using concept lattices for text retrieval and mining. In B. Ganter, G. Stumme & R. Wille, eds., *Formal Concept Analysis*, vol. 3626 of *Lecture Notes in Computer Science*, 161–179, Springer Berlin Heidelberg. 64

CHEIN, M. & MUGNIER, M.L. (1992). Conceptual Graphs: fundamental notions. *Revue d'Intelligence Artificielle*, **6**, 365–406. 51, 56

CHEUNG, K.S. & VOGEL, D. (2005). Complexity reduction in lattice-based information retrieval. *Inf. Retr.*, **8**, 285–299. 64

CHEVALLET, J.P. (2009). endogènes et exogènes pour une indexation conceptuelle intermédia. Mémoire d'Habilitation a Diriger des Recherches. 21, 22

CHEVALLET, J.P. & CHIARAMELLA, Y. (1995). Extending a logic-based retrieval model with algebraic knowledge. In *MIRO*. 8, 9, 10, 55, 59, 65

CHEVALLET, J.P. & CHIARAMELLA, Y. (1998). Experiences in information retrieval modelling using structured formalisms and modal logic. In F. Crestani, M. Lalmas & C. Rijsbergen, eds., *Information Retrieval: Uncertainty and Logics*, vol. 4 of *The Kluwer International Series on Information Retrieval*, 39–72, Springer US. 55, 70, 84

CHEVALLET, J.P., LIM, J.H. & LE, D.T.H. (2007). Domain knowledge conceptual intermedia indexing: application to multilingual multimedia medical reports. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, CIKM '07, 495–504, ACM, New York, NY, USA. 7, 18, 19, 20, 22, 23, 24

CHIARAMELLA, Y. & CHEVALLET, J.P. (1992). About retrieval models and logic. *Comput. J.*, **35**, 233–242. 6, 10, 20, 70, 72, 73, 84, 86, 108

CHIARAMELLA, Y., MULHEM, P. & FOUREL, F. (1996). A model for multimedia information retrieval. 60

CLINCHANT, S. & GAUSSIER, E. (2010). Information-based models for ad hoc ir. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '10, 234–241, ACM, New York, NY, USA. 123, 130, 138, 160

CLINCHANT, S. & GAUSSIER, E. (2013). A theoretical analysis of pseudo-relevance feedback models. In *Proceedings of the 2013 Conference on the Theory of Information Retrieval*, ICTIR '13, 6:6–6:13, ACM, New York, NY, USA. 160

CODOCEDO, V., LYKOURENTZOU, I. & NAPOLI, A. (2012). A contribution to semantic indexing and retrieval based on fca - an application to song datasets. In *CLA*, 257–268. 20, 64

COLLINS-THOMPSON, K. & CALLAN, J. (2005). Query expansion using random walk models. In *Proceedings of the 14th ACM International Conference on Information and Knowledge Management*, CIKM '05, 704–711, ACM, New York, NY, USA. 30

COX, R.T. (1946). Probability, Frequency and Reasonable Expectation. *American Journal of Physics*, **14**, 1–13. 11, 178

CRESTANI, F. (1998). Logical imaging and probabilistic information retrieval. In F. Crestani, M. Lalmas & C. Rijsbergen, eds., *Information Retrieval: Uncertainty and Logics*, vol. 4 of *The Kluwer International Series on Information Retrieval*, 247–279, Springer US. 48, 49

CRESTANI, F. (2000). Exploiting the similarity of non-matching terms at retrievaltime. *Inf. Retr.*, **2**, 27–47. 7, 20, 21, 60

CRESTANI, F. & LALMAS, M. (2001). Logic and uncertainty in information retrieval. In *Proceedings of the Third European Summer-School on Lectures on Information Retrieval-Revised Lectures*, ESSIR '00, 179–206, Springer-Verlag, London, UK, UK. 70, 84

CRESTANI, F. & RIJSBERGEN, C.J.V. (1995). Information retrieval by logical imaging. *Journal of Documentation*, **51**, 3–17. 8, 9, 48, 49, 50, 59, 60, 65, 152

CRESTANI, F., LALMAS, M. & RIJSBERGEN, C. (1998). *Information retrieval: uncertainty and logics : advanced models for the representation and retrieval of information*. The Kluwer international series on information retrieval, Kluwer Academic Publishers. 6

CROFT, W.B. (2000). *Combining Approaches to Information Retrieval*, vol. 7. Springer. 24

CUI, H., WEN, J.R., NIE, J.Y. & MA, W.Y. (2002). Probabilistic query expansion using query logs. In *Proceedings of the 11th International Conference on World Wide Web*, WWW '02, 325–332, ACM, New York, NY, USA. 30

CUI, H., WEN, J.R., NIE, J.Y. & MA, W.Y. (2003). Query expansion by mining user logs. *IEEE Trans. on Knowl. and Data Eng.*, **15**, 829–839. 30

DAS-GUPTA, P. & KATZER, J. (1983). A study of the overlap among document representations. *SIGIR Forum*, **17**, 106–114. 24

DAVIS, R., SHROBE, H.E. & SZOLOVITS, P. (1993). What is a knowledge representation? *AI Magazine*, **14**, 17–33. 17

DEERWESTER, S. (1988). Improving information retrieval with latent semantic indexing. In C.L. Borgman & E.Y.H. Pai, eds., *Proceedings of the 51st ASIS Annual Meeting (ASIS '88)*, vol. 25, American Society for Information Science, Atlanta, Georgia. 22

DEERWESTER, S., DUMAIS, S.T., FURNAS, G.W., LANDAUER, T.K. & HARSHMAN, R. (1990). Indexing by latent semantic analysis. *JOURNAL OF THE AMERICAN SOCIETY FOR INFORMATION SCIENCE*, **41**, 391–407. 22

DIAZ, F. & METZLER, D. (2006). Improving the estimation of relevance models using large external corpora. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '06, 154–161, ACM, New York, NY, USA. 30

DIEM, L.T.H., CHEVALLET, J.P. & THUY, D.T.B. (2007). Thesaurus-based query and document expansion in conceptual indexing with umls: Application in medical information retrieval. In *Research, Innovation and Vision for the Future, 2007 IEEE International Conference on*, 242–246. 30

DINH, D. (2012). *Accès à l'information biomédicale : vers une approche d'indexation et de recherche d'information conceptuelle basée sur la fusion de ressources termino-ontologiques*. Thèse de doctorat, Université Paul Sabatier, Toulouse, France. 128

DJOUADI, Y. (2012). Généralisation des opérateurs de dérivation de galois en recherche d'information basée sur l'analyse formelle de concepts. In *CORIA*, 373–386. 64

DOZIER, C., KONDADADI, R., AL-KOFAHI, K., CHAUDHARY, M. & GUO, X. (2007). Fast tagging of medical terms in legal text. In *Proceedings of the 11th international conference on Artificial intelligence and law*, ICAIL '07, 253–260, ACM, New York, NY, USA. 23, 25, 28,

29, 120

EFTHIMIADIS, E.N. (1996). Query expansion. *Annual Review of Information Systems and Technology (ARIST)*, **31**, 121–187. 21

FANG, H. & ZHAI, C. (2005). An exploration of axiomatic approaches to information retrieval. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '05, 480–487, ACM, New York, NY, USA. 123, 129

FANG, H. & ZHAI, C. (2006). Semantic term matching in axiomatic approaches to information retrieval. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '06, 115–122, ACM, New York, NY, USA. 30, 158

FANG, H., TAO, T. & ZHAI, C. (2004). A formal study of information retrieval heuristics. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '04, 49–56, ACM, New York, NY, USA. 104, 123, 129, 138, 160

FOX, E.A. & SHAW, J.A. (1994). Combination of multiple searches. In *The Second Text REtrieval Conference (TREC-2)*, vol. 500-215 of *NIST Special Publication*, 243–252, NIST. 24

FRAKES, W.B. (1992). *Stemming algorithms*, 131–160. Prentice-Hall, Inc., Upper Saddle River, NJ, USA. 22

FUHR, N. (1995). Probabilistic datalog - a logic for powerful retrieval methods. In *Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '95, 282–290, ACM, New York, NY, USA. 57, 59

GARDENFORS, P. (1982). Imaging and Conditionalization. *The Journal of Philosophy*, **79**, 747–760. 46, 47

GOBEILL, J., RUCH, P. & ZHOU, X. (2009). Query and document expansion with medical subject headings terms at medical imageclef 2008. In C. Peters, T. Deselaers, N. Ferro, J. Gonzalo, G. Jones, M. Kurimo, T. Mandl, A. Peñas & V. Petras, eds., *Evaluating Systems for Multilingual and Multimodal Information Access*, vol. 5706 of *Lecture Notes in Computer Science*, 736–743, Springer Berlin Heidelberg. 30

GONZALO, J., VERDEJO, F., CHUGUR, I. & CIGARRAN, J. (1998). Indexing with WordNet synsets can improve text retrieval. *Proceedings of the COLING/ACL Workshop on Usage of WordNet in Natural Language Processing Systems*, 38–44. 30

GREFENSTETTE, G. (1992). Use of syntactic context to produce term association lists for text retrieval. In *Proceedings of the 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '92, 89–97, ACM, New York, NY, USA. 5

HIEMSTRA, D. & DE VRIES, A.P. (2000). Relating the new language models of information retrieval to the traditional retrieval models. 94

HO, B.Q., THUY, D.T.B., CHEVALLET, J.P. & BRUANDET, M. (2006). A structured indexing model based on noun phrases. In *Research, Innovation and Vision for the Future, 2006 International Conference on*, 81–89. 21

HOLI, M. & HYVÖNEN, E. (2005). Modeling degrees of conceptual overlap in semantic web ontologies. In P.C.G. da Costa, K.B. Laskey, K.J. Laskey & M. Pool, eds., *ISWC-URSW*, 98–99. 22, 122

HUIBERS, T. & BRUZA, P. (1994). *Situations: A General Framework for Studying Information Retrieval*. Technical report (Rijksuniversiteit te Utrecht. Dept. of Computer Science), Utrecht

University, Department of Computer Science. 56, 65

HUNTER, A. (1995). Using default logic in information retrieval. In *Symbolic and Quantitative Approaches to Uncertainty, volume 946 of Lecture Notes in Computer Science*, 235–242, Springer. 9, 58, 59, 65

JAEGER, M. (1994). Probabilistic reasoning in terminological logics. In *Proceedings of KR-94*, 305–316. 54

JAEGER, M. (2006). Probabilistic role models and the guarded fragment. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, **14**, 43–60. 54

JONES, K.S. (1972). A Statistical Interpretation of Term Specificity and its Application in Aetrieval. *Journal of Documentation*, **28**, 11–21. 58

KIM, Y., SEO, J. & CROFT, W.B. (2011). Automatic boolean query suggestion for professional search. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '11, 825–834, ACM, New York, NY, USA. 116

KNUTH, K.H. (2003). Deriving laws from ordering relations. In *In press: Bayesian Inference and Maximum Entropy Methods in Science and Engineering, Jackson Hole WY*, 204–235. 11, 78, 178

KNUTH, K.H. (2005). Lattice duality: The origin of probability and entropy. *Neurocomput.*, **67**, 245–274. 10, 11, 77, 78, 86, 91, 153, 177, 178

KOLLER, D., LEVY, A. & PFEFFER, A. (1997). P-classic: A tractable probabilistic description logic. In *Proceedings of AAAI-97*, 390–397. 51, 65

KRIPKE, S.A. (1963). Semantic analysis of modal logic I: Normal modal and propositional calculi. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik*, **9**, 67–96. 46, 48, 55

LALMAS, M. (1998). Logical models in information retrieval: Introduction and overview. In *Information Processing & Management*, 34–1. 6

LALMAS, M. & BRUZA, P.D. (1998). The use of logic in information retrieval modelling. *The Knowledge Engineering Review*, **13**, 263–295. 57, 81

LALMAS, M. & RIJSBERGEN, K. (1993). A logical model of information retrieval based on situation theory. In T. McEnery & C. Paice, eds., *14th Information Retrieval Colloquium*, Workshops in Computing, 1–13, Springer London. 56, 59, 65

LAVRENKO, V. & CROFT, W.B. (2001). Relevance based language models. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '01, 120–127, ACM, New York, NY, USA. 94

LE, T.H.D. (2009). *Utilisation de ressource externes dans un modèle Bayesien de Recherche d'Information: Application a la recherche d'information médicale multilingue avec UMLS*. Ph.D. thesis, Université Joseph Fourier, Ecole Doctorale MSTII. 24, 31, 35, 36, 37

LEACOCK, C. & CHODOROW, M. (1998). *Combining local context and WordNet similarity for word sense identification*, 305–332. In C. Fellbaum (Ed.), MIT Press. 32, 36, 122

LEE, J.H. (1995). Combining multiple evidence from different properties of weighting schemes. In *Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '95, 180–188, ACM, New York, NY, USA. 24

LEWIS, D.K. (1973). *Counterfactuals*. Harvard University Press. 46

LI, Y., BANDAR, Z.A. & MCLEAN, D. (2003). An approach for measuring semantic similarity between words using multiple information sources. *IEEE Trans. on Knowl. and Data Eng.*, **15**, 871–882. 22, 122

LI, Y., LUK, W.P.R., HO, K.S.E. & CHUNG, F.L.K. (2007). Improving weak ad-hoc queries using wikipedia asexternal corpus. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '07, 797–798, ACM, New York, NY, USA. 30

LIN, D. (1998). An information-theoretic definition of similarity. In *Proceedings of the Fifteenth International Conference on Machine Learning*, ICML '98, 296–304, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA. 32

LOSADA, D.E. & BARREIRO, A. (2001). A logical model for information retrieval based on propositional logic and belief revision. *Comput. J.*, **44**, 410–424. 8, 9, 43, 59, 60, 70, 99, 111, 152

LOSADA, D.E. & BARREIRO, A. (2003). Propositional logic representations for documents and queries: a large-scale evaluation. In *Proceedings of the 25th European conference on IR research*, ECIR'03, 219–234, Springer-Verlag, Berlin, Heidelberg. 45, 116

LUHN, H.P. (1958). The automatic creation of literature abstracts. *IBM J. Res. Dev.*, **2**, 159–165. 123, 138, 166

LUKASIEWICZ, T. (2008). Expressive probabilistic description logics. *Artif. Intell.*, **172**, 852–883. 51, 53

MAISONNASSE, L. (2008). *Les supports de vocabulaires pour les systèmes de recherche d'information orientés précision : application aux graphes pour la recherche d'information médicale.*. Ph.D. thesis, Université Joseph-Fourier - Grenoble I. 24, 25, 28, 31, 33, 37, 120

MAISONNASSE, L., GAUSSIER, E. & CHEVALLET, J.P. (2009). Model fusion in conceptual language modeling. In *Proceedings of the 31th European Conference on IR Research on Advances in Information Retrieval*, ECIR '09, 240–251, Springer-Verlag, Berlin, Heidelberg. 7, 23, 25, 28, 33, 103, 120

MANDALA, R., TOKUNAGA, T. & TANAKA, H. (1999). Combining multiple evidence from different types of thesaurus for query expansion. In *Proceedings of the 22Nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '99, 191–197, ACM, New York, NY, USA. 30

MEGHINI, C., SEBASTIANI, F., STRACCIA, U. & THANOS, C. (1993). 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, 298–307, ACM, New York, NY, USA. 7, 9, 17, 20, 39, 52, 53, 59, 70, 84

MESSAI, N., DEVIGNES, M.D., NAPOLI, A. & SMAÏL-TABBONE, M. (2006). BR-Explorer: An FCA-based algorithm for Information Retrieval. In *Fourth International Conference On Concept Lattices and Their Applications - CLA 2006*, Hammamet/Tunisia. 63

METZLER, D. & CROFT, W.B. (2004). Combining the language model and inference network approaches to retrieval. *Inf. Process. Manage.*, **40**, 735–750. 98

MOHLER, M. & MIHALCEA, R. (2009). Text-to-text semantic similarity for automatic short answer grading. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, EACL '09, 567–575, Association for Computational Linguistics, Stroudsburg, PA, USA. 22, 122

MOOERS, C. (1958). *A Mathematical Theory of Language Symbols in Retrieval: Paper Proposed to International Confernece on Scientific Information, Washington, D.C., November 1958*. Zator Company. 8, 61, 70

NAVIGLI, R. (2009). Word sense disambiguation: A survey. *ACM Comput. Surv.*, **41**, 10:1–10:69. 23

NIE, J. (1988). An outline of a general model for information retrieval systems. In *Proceedings*

of the 11th annual international ACM SIGIR conference on Research and development in information retrieval, SIGIR '88, 495–506, ACM, New York, NY, USA. 8, 9, 48, 49, 50, 56, 59, 65, 69, 70, 81, 84, 87, 90, 99

NIE, J. (1989). An information retrieval model based on modal logic. *Information Processing & Management*, **25**, 477 – 491. 48, 49, 50, 59

NIE, J.Y. (1992). 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, 140–151, ACM, New York, NY, USA. 49, 50, 59, 65

NIE, J.Y. & BRISEBOIS, M. (1996). An inferential approach to information retrieval and its implementation using a manual thesaurus. *Artif. Intell. Rev.*, **10**, 409–439. 5, 7, 8, 30, 39, 50, 59

OUNIS, I. & HUIBERS, T. (1997). A logical relational approach for information retrieval indexing. In *Proceedings of the 19th Annual BCS-IRSG conference on Information Retrieval Research*, IRSG'97, 7–7, British Computer Society, Swinton, UK, UK. 17

PICARD, J. & SAVOY, J. (2000). A logical information retrieval model based on a combination of propositional logic and probability theory. In F. Crestani & G. Pasi, eds., *Soft Computing in Information Retrieval*, vol. 50 of *Studies in Fuzziness and Soft Computing*, 225–258, Physica-Verlag HD. 8, 45, 59, 152

PONTE, J.M. & CROFT, W.B. (1998). A language modeling approach to information retrieval. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '98, 275–281, ACM, New York, NY, USA. 11, 31, 92, 129, 138

PORTER, M.F. (1997). An algorithm for suffix stripping. In K. Sparck Jones & P. Willett, eds., *Readings in Information Retrieval*, 313–316, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA. 120

PRISS, U. (2000). Lattice-based information retrieval. *KNOWLEDGE ORGANIZATION*, **27**, 132–142. 61

QI, G. & PAN, J.Z. (2008). A tableau algorithm for possibilistic description logic $\mathcal{ALC}$. In *Proceedings of the 3rd Asian Semantic Web Conference on The Semantic Web*, ASWC '08, 61–75, Springer-Verlag, Berlin, Heidelberg. 51, 54

QIU, Y. & FREI, H.P. (1993). Concept based query expansion. In *Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '93, 160–169, ACM, New York, NY, USA. 21, 22, 30

RADHOUANI, S. (2008). *Un modèle de recherche d'information orienté précision fondé sur les dimensions de domaine*. Ph.D. thesis, Co-tutelle Université Joseph Fourier Grenoble, Université de Genève (Suisse). 24

RAJAPAKSE, R.K. & DENHAM, M. (2006). Text retrieval with more realistic concept matching and reinforcement learning. *Inf. Process. Manage.*, **42**, 1260–1275. 64

REN, F. & BRACEWELL, D.B. (2009). Advanced information retrieval. *Electron. Notes Theor. Comput. Sci.*, **225**, 303–317. 20

RESNIK, P. (1999). Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language. *J. Artif. Intell. Res. (JAIR)*, **11**, 95–130. 32

ROBERTSON, S.E. (1977). The Probability Ranking Principle in IR. *Journal of Documentation*, **33**, 294–304. 31, 93

ROBERTSON, S.E. & JONES, K.S. (1976). Relevance weighting of search terms. *J. Am. Soc.*

Inf. Sci., **27**, 129–146. 94

ROBERTSON, S.E. & WALKER, S. (1994). Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '94, 232–241, Springer-Verlag New York, Inc., New York, NY, USA. 129, 138

ROCCHIO, J.J. (1971). Relevance feedback in information retrieval. In G. Salton, ed., *The Smart retrieval system - experiments in automatic document processing*, 313–323, Englewood Cliffs, NJ: Prentice-Hall. 21

ROUSSEY, C. (2001). *Une Méthode d'indexation sémantique adaptée aux corpus multilingues*. Ph.D. thesis, INSA, thèse de l'INSA de Lyon, 2001. 31

SALTON, G. (1971). *The SMART Retrieval System—Experiments in Automatic Document Processing*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA. 120

SALTON, G. & BUCKLEY, C. (1997). Improving retrieval performance by relevance feedback. In K. Sparck Jones & P. Willett, eds., *Readings in information retrieval*, 355–364, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA. 21

SALTON, G., WONG, A. & YANG, C.S. (1975). A vector space model for automatic indexing. *Commun. ACM*, **18**, 613–620. 31, 45, 96

SANDERSON, M. (1994). Word sense disambiguation and information retrieval. In *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '94, 142–151, Springer-Verlag New York, Inc., New York, NY, USA. 23, 113, 114

SEBASTIANI, F. (1994). 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, 122–130, Springer-Verlag New York, Inc., New York, NY, USA. 53, 54, 59

SEBASTIANI, F. (1998). On the role of logic in information retrieval. In *Information Processing and Management*, 1–18. 41, 69, 71, 73, 74

SEBASTIANI, F. & STRACCIA, U. (1991). A computationally tractable terminological logic. In *SCAI*, 307–315. 51, 65

SHAW, J.A., FOX, E.A., SHAW, J.A. & FOX, E.A. (1994). Combination of multiple searches. In *The Second Text REtrieval Conference (TREC-2*, 243–252. 24

SINGHAL, A. (2001). Modern Information Retrieval: A Brief Overview. *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering*, **24**, 35–42. 106, 129

SINGHAL, A., BUCKLEY, C. & MITRA, M. (1996). Pivoted document length normalization. In *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '96, 21–29, ACM, New York, NY, USA. 129, 138

SMUCKER, M.D., ALLAN, J. & CARTERETTE, B. (2007). A comparison of statistical significance tests for information retrieval evaluation. In *Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management*, CIKM '07, 623–632, ACM, New York, NY, USA. 128

SOWA, J.F. (1984). *Conceptual structures: information processing in mind and machine*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA. 54

STYLTSVIG, H. (2006). *Ontology-based Information Retrieval*. Datalogiske Skrifter, Roskilde Universitet. 31

TURTLE, H. & CROFT, W.B. (1990). Inference networks for document retrieval. In *Proceedings of the 13th annual international ACM SIGIR conference on Research and development*

in information retrieval, SIGIR '90, 1–24, ACM, New York, NY, USA. 97

TURTLE, H. & CROFT, W.B. (1991). Evaluation of an inference network-based retrieval model. *ACM Trans. Inf. Syst.*, **9**, 187–222. 97

VALLET, D., FERNÁNDEZ, M. & CASTELLS, P. (2005). An ontology-based information retrieval model. In A. Gómez-Pérez & J. Euzenat, eds., *The Semantic Web: Research and Applications*, vol. 3532 of *Lecture Notes in Computer Science*, 455–470, Springer Berlin Heidelberg. 30

VAN RIJSBERGEN, C.J. (1977). A theoretical basis for the use of co-occurrence data in information retrieval. *Journal of Documentation*, **33**, 106–119. 5, 50

VAN RIJSBERGEN, C.J. (1986). A non-classical logic for information retrieval. *Comput. J.*, **29**, 481–485. 4, 9, 20, 39, 40, 42, 48, 65, 69, 70, 72, 73, 81, 84, 89, 90

VAN RIJSBERGEN, C.J. (2004). *The geometry of information retrieval*. Cambridge University Press. 157

VOGT, C.C. & COTTRELL, G.W. (1998). Predicting the performance of linearly combined ir systems. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '98, 190–196, ACM, New York, NY, USA. 24

VOGT, C.C. & COTTRELL, G.W. (1999). Fusion via a linear combination of scores. *Inf. Retr.*, **1**, 151–173. 24

VOORHEES, E.M. (1993). On expanding query vectors with lexically related words. In *TREC*, 223–232. 30

VOORHEES, E.M. (1994). Query expansion using lexical-semantic relations. In *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '94, 61–69, Springer-Verlag New York, Inc., New York, NY, USA. 30

VOORHEES, E.M. & HARMAN, D. (1999). Overview of the Eighth Text REtrieval Conference (TREC-8). In *TREC*. 136

VOORHEES, E.M. & HARMAN, D. (2000). Overview of the Sixth Text REtrieval Conference (TREC-6). *Inf. Process. Manage.*, **36**, 3–35. 135

WILLE, R. (1982). Restructuring Lattice Theory: an Approach Based on Hierarchies of Concepts. In I. Rival, ed., *Ordered sets*, 445–470, Reidel, Dordrecht–Boston. 20, 62

WILLE, R. (2005). Formal concept analysis as mathematical theory of concepts and concept hierarchies. In B. Ganter, G. Stumme & R. Wille, eds., *Formal Concept Analysis*, 1–33, Springer-Verlag, Berlin, Heidelberg. 62

WOLFF, K.E. (1993). A first course in Formal Concept Analysis. In F. Faulbaum, ed., *StatSoft '93*, 429–438, Gustav Fischer Verlag. 62

WOODS, W. (1997). Conceptual indexing: A better way to organize knowledge. 21

YIN, Z., SHOKOUHI, M. & CRASWELL, N. (2009). Query expansion using external evidence. In M. Boughanem, C. Berrut, J. Mothe & C. Soule-Dupuy, eds., *Advances in Information Retrieval*, vol. 5478 of *Lecture Notes in Computer Science*, 362–374, Springer Berlin Heidelberg. 30

ZHAI, C. (2008). *Statistical Language Models for Information Retrieval*. Synthesis Lectures on Human Language Technologies, Morgan & Claypool Publishers. 94, 129

ZHAI, C. & LAFFERTY, J. (2001). A study of smoothing methods for language models applied to ad hoc information retrieval. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '01, 334–342, ACM, New York, NY, USA. 129, 130, 138

ZHAO, J., BOLEY, H. & DU, W. (2012). A fuzzy logic based approach to expressing and reasoning with uncertain knowledge on the semantic web. In K. Madani, A. Dourado Correia, A. Rosa & J. Filipe, eds., *Computational Intelligence*, vol. 399 of *Studies in Computational Intelligence*, 167–181, Springer Berlin Heidelberg. 54

ZHOU, X., ZHANG, X. & HU, X. (2006). Using concept-based indexing to improve language modeling approach to genomic ir. In M. Lalmas, A. MacFarlane, S. Rüger, A. Tombros, T. Tsikrika & A. Yavlinsky, eds., *Advances in Information Retrieval*, vol. 3936 of *Lecture Notes in Computer Science*, 444–455, Springer Berlin Heidelberg. 30

ZUCCON, G., AZZOPARDI, L. & VAN RIJSBERGEN, C.J. (2009). Revisiting logical imaging for information retrieval. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '09, 766–767, ACM, New York, NY, USA. 50