

International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems
© World Scientific Publishing Company

ON AN INTERPRETATION OF KEYWORDS WEIGHTS IN INFORMATION RETRIEVAL: SOME FUZZY LOGIC BASED APPROACHES

ŚLAWOMIR ZADROŹNY

*Systems Research Institute, Polish Academy of Sciences, ul. Newelska 6
01-416 Warszawa, Poland
Slawomir.Zadrozny@ibspan.waw.pl*

JANUSZ KACPRZYK

*Systems Research Institute, Polish Academy of Sciences, ul. Newelska 6
01-416 Warszawa, Poland
Janusz.Kacprzyk@ibspan.waw.pl*

Received (received date)

Revised (revised date)

Relevant contributions of fuzzy logic to the logical models in information retrieval is studied. It makes it possible to grasp the graduality of some relevant concepts and to model both imprecision and uncertainty inherent to the retrieval process, still in the framework of the broadly meant logical approach. In this perspective we discuss various extensions to the basic Boolean model which are needed to attain such a greater expressivity. In particular, we show how the well-known semantics of keywords weights may be recovered in various fuzzy logic based information retrieval models.

Keywords: information retrieval, fuzzy logic, keywords weights, semantics, imprecision, uncertainty.

1. Introduction

Logical models in information retrieval (IR) provide a rich and theoretically well-founded formalism to represent documents and queries, and to define their matching in this framework.¹ The classical binary logic has some difficulties in dealing with many concepts crucial for information retrieval, such as the *relevance* of a document against a query or the *importance* of a keyword for the representation of a document/query. This is due to their inherently gradual nature: a document may be relevant *to a degree*, more or less so than other document. The same applies to the importance of keywords. Fuzzy logic has been quickly recognized as a convenient formal modelling tool to tackle this problems in the area of information retrieval.

In fact research on the application of fuzzy logic in IR has a long history and has led to the development of many interesting approaches. However, even if they belong to the family of the logical information retrieval models they usually do not offer

2 *S. Zadrozny, J. Kacprzyk*

such a strict logical formalism as expected. This paper is an attempt to present some known fuzzy logic based approaches in a more formal way, typical for logical models. Different approaches under consideration are based on various understanding of what fuzzy logic is. We briefly discuss a model we recently proposed and which employs fuzzy logic in a so-called broad sense, to be more specific in the sense of Zadeh's calculus of linguistic statements.

Most of the classical IR models, including all discussed here, adopt the keyword-based representation of the documents and queries. The basic logical model ("the Boolean model") admits only the binary importance weights for the keywords, i.e., a keyword may only be declared as (completely) important or (completely) unimportant. However, it is a widely accepted belief that distinguishing the weights of keywords leads to a higher effectiveness of the retrieval. Thus, many proposed extensions of the Boolean model, including those fuzzy logic based, have made attempts to overcome limitations in this respect. Allowing non-binary keywords weights in queries and documents raises a natural question of their semantics.² In this paper we address this issue showing a formal representation of various semantics in the framework of fuzzy logic based approaches. Thus, our aim is to provide a comprehensive and consistent logical treatment of known approaches of this type.

2. The Boolean Model – A Starting Point for Fuzzy Extensions

Fuzzy logic based models of information retrieval may be seen as emerging from the basic logical model, usually referred to as the Boolean model.³ Here we briefly recall the main points of this model. Three major components of any information retrieval model comprise the logical representation of documents and queries as well as the way their matching is defined. The Boolean model may conveniently be defined in the following way.

We will use the following notation:

$D = \{d_i\}_{i \in [1, N]}$ – is a set of documents

q – is a query

$T = \{t_j\}_{j \in [1, M]}$ – is a set of index terms; a vocabulary

A propositional variable $s_j \in S$ is associated with each keyword $t_j \in T$, $S = \{s_j\}_{j \in [1, M]}$.

Documents and queries are represented by some formulas of propositional logic over the alphabet S . We will denote the formulas representing a document d and a query q by ϕ_d and ϕ_q , respectively. A query is a formula stating which combinations of keywords should appear (or should not appear) in the documents sought. Similarly, a document is also represented by a formula indicating which (combinations of) keywords are important to express its meaning. A typical representation takes the form of a conjunction

$$s_1 \wedge s_2 \wedge s_3 \tag{1}$$

where s_i corresponds to an index term t_i found to be important for the representation of this document. Depending on additional assumptions, the variables/index terms not explicitly used in such a conjunction are implicitly treated as negated (under the Closed World Assumption, or CWA for short) or not (under the Open World Assumption, or OWA for short). In general, a formula representing a document may take on a more complex form involving propositional variables and logical connectives of negation, conjunction and disjunction. The same applies to the query representation. It may be a compound formula but often is just a conjunction what is exemplified by the interfaces of search engines and their default way of treating sequences of keywords constituting a query.

For a further discussion it is convenient to identify a document d with the model (in the CWA case) or the set of models (in the OWA case) $\Omega_d = \{\omega_d\}$ of the formula ϕ_d , i.e., set of such valuations $\omega : S \rightarrow \{0, 1\}$ of the propositional variables in S that make the formula ϕ_d true:

$$d \mapsto \Omega^d = \{\omega \mid \omega : S \rightarrow \{0, 1\}, \omega(\phi_d) = 1\} \quad (2)$$

The matching between a query and a document may be then formally defined as follows. A document d matches a query q (d is deemed to be relevant with respect to q) if ϕ_q is true under all valuations $\omega_d \in \Omega_d$.

Such a basic logical approach offers a highly expressive language to represent documents and queries. However in the framework of the classical logic it is not possible to directly represent either a varying *importance* of the keywords for the representation of documents/queries or a *gradual* character of the matching. The use of fuzzy logic helps preserve the benefits of the logical approach while overcoming these limitations.

In the literature many extensions of such a basic Boolean model have been proposed. They jointly form the class of *logical models*. They may be briefly characterized as assuming the representation of queries and documents using some notions of a selected logic. A comprehensive survey of the relevant approaches may be found in papers by Lalmas¹ and by Crestani and Lalmas⁴. We are here in particular interested in approaches where queries and documents are represented by formulas of given logic. The matching between a query represented by a formula q and a document represented by a formula d may be then defined, as in Rijsbergen's approach^{5,6}, as a form of a logical dependency between these formulas, which is denoted as

$$d \Rightarrow q \quad (3)$$

This dependency may be meant in many various ways, cf. ⁷. for a discussion.

3. Fuzzy Propositional Logic and its Variants – A Tool to Extend the Boolean Model

One of natural ways to make the classical Boolean model more flexible is via the use of a fuzzy logic in a so-called narrow sense, i.e., meant as a variant of the

4 *S. Zadrozny, J. Kacprzyk*

multivalued logic. Another view of fuzzy logic is the one in a so-called broad sense which is basically meant as a basis of approximate reasoning schemes.

Here we will briefly discuss the basic principles of a *fuzzy propositional logic*. Depending on the scope of the sought extension to the classical Boolean model a more or less elaborated version of such a logic is needed. We start with its simple version, traditionally related to fuzzy sets theory. We focus on the semantic aspects of such a logic.

The language of this simple version of fuzzy logic is identical with the classical one. Semantics is based on the structure:

$$\mathcal{L} = ([0, 1], \max, \min, \neg, \rightarrow) \quad (4)$$

where particular operators correspond to the disjunction, conjunction, negation and implication, respectively. The negation and implication operators are usually assumed to be $\neg x = 1 - x$ and $x \rightarrow y = \max(1 - x, y)$, respectively. Other forms of the operators are also possible. Now, a valuation ω assigns to each propositional variable a number from the interval $[0, 1]$ and compound formulas are valued employing the indicated correspondence between the logical connectives and the operators listed in the structure \mathcal{L} , cf. (4).

Such a fuzzy propositional logic is sufficient for a simple, straightforward extension of the Boolean model (cf. next section). However, a more comprehensive extension requires a more sophisticated version of the fuzzy propositional logic. The Rational Pavelka logic seems to be a perfect candidate (cf., Hajek's presentation in ⁸ as well as the works of Perfilieva and Novak ^{9,10}). In this the structure of truth values have the form of a complete residuated lattice:

$$\mathcal{L} = (L, \vee, \wedge, \otimes, \rightarrow, \mathbf{0}, \mathbf{1}) \quad (5)$$

which is a generalization of the conventional Boolean algebra used in the classical logic. It is equipped with four binary operations: two lattice operations \wedge and \vee , the multiplication (\otimes) and the residuation (\rightarrow).

Basically, J , an extended formal language (alphabet) of the classical propositional logic, and a set, \mathcal{F}_J , of well-formed formulas over it, are assumed as a point of departure. The alphabet J consists of:

- a countable set of propositional variables s_1, s_2, \dots
- a set of logical constants $\{\mathbf{a} \mid a \in L\}$ (including \perp and \top corresponding to $\mathbf{0}$ and $\mathbf{1}$, respectively)
- the symbol of the logical connective of implication \Rightarrow
- brackets as auxiliary symbols.

The well-formed formulas over this alphabet are defined as in the classical propositional logic (cf. the discussion of the role of the logical constants, given in what follows).

Other logical connectives in addition to the implication may be derived in the usual way. An important extension consists in adding to the alphabet logical con-

stants for all the truth values of L . They, themselves, will be treated as the elementary formulas (besides the classical elementary formulas composed of a single propositional variable). Moreover, the concept of an evaluated formula is introduced as a pair (p, a) of a well-formed formula p and its syntactic evaluation a . The intuition behind the concept of an evaluated formula is such that it requires a formula p to be true to a degree equal at least a . In fact, such an evaluated formula may be interpreted as a regular formula $\mathbf{a} \Rightarrow p$. This interpretation will be very useful for our purposes.

We skip other proof-theoretical aspects of this logic (inference rules, logical axioms) as we are mainly interested in issues related to semantics.

A valuation ω in this logic is defined similarly as in the classical case:

$$\omega : \mathcal{F}_J \longrightarrow L \quad (6)$$

For an elementary formula a value is assigned directly and for a compound one by employing the truth-functionality of the logical connectives. In particular:

$$\omega(p \Rightarrow q) = \omega(p) \rightarrow \omega(q) \quad (7)$$

$$\omega(p \vee q) = \omega(p) \vee \omega(q) \quad (8)$$

$$\omega(p \wedge q) = \omega(p) \wedge \omega(q) \quad (9)$$

$$\omega(\mathbf{a}) = a \quad (10)$$

The truth values come now from $[0, 1]$ (in general, from a complete residuated lattice L) instead of $\{0, 1\}$. New, special atomic formulas built of logical constants alone, \mathbf{a} , are always assigned a truth value a (i.e. a truth value which the logical constant \mathbf{a} corresponds to).

Still another logic is useful for our consideration, namely the *possibilistic logic*¹¹. In what follows we briefly recall some basic features of its propositional calculus version. Let us start again with the classical propositional logic and consider, as previously, the set of propositional variables $S = \{s_j\}$ and denote by $\Omega = \{\omega\}$ the set of all their classical binary valuations $\omega : S \rightarrow \{0, 1\}$. \mathcal{F} is the set of all formulas built following the usual well-formedness rules. The valuations of Ω are in a standard way extended on \mathcal{F} . For a formula $\phi \in \mathcal{F}$ let Ω^ϕ denote the set of all its models: $\Omega^\phi = \{\omega \in \Omega : \omega(\phi) = 1\}$.

A formula $\phi \in \mathcal{F}$ may be seen as inducing a possibility distribution π_ϕ ¹² on the set Ω such that

$$\pi_\phi(\omega) = \begin{cases} 1 & \forall \omega \in \Omega^\phi \\ 0 & \forall \omega \notin \Omega^\phi \end{cases} \quad (11)$$

Then assuming that the knowledge of an agent is expressed by a formula ϕ (or, more precisely, assuming the agent's total belief in the truth of ϕ), the belief of this agent in the truth of any other formula $\psi \in \mathcal{F}$ is expressed by a pair of numbers:

$$(\Pi_\phi(\psi), N_\phi(\psi)) = (\Pi_\phi(\Omega^\psi), N_\phi(\Omega^\psi)) \quad \Pi_\phi(\psi), N_\phi(\psi) \in \{0, 1\} \quad (12)$$

6 *S. Zadrozny, J. Kacprzyk*

where Π_ϕ and N_ϕ denote the possibility and necessity measures with respect to the possibility distribution π_ϕ given by (11). The pair of numbers $(1, 1)$ denotes the total certainty that formula ψ is true ($\neg\psi$ is false), the pair $(0, 0)$ denotes the total certainty that formula ψ is false ($\neg\psi$ is true), and the pair $(1, 0)$ denotes the case where both ψ and $\neg\psi$ may be true or false. Due to the binary character of the possibility distribution π_ϕ also the values of the measures of possibility and necessity Π_ϕ and N_ϕ belong to the set $\{0, 1\}$.

The possibilistic logic provides a means to model an agent's limited belief in the truth of a formula. It should be stressed that the truth of the formula is still binary, i.e., it may be only true or false, but the belief in its truth may be expressed in the gradual way. Such a limited belief in the truth of a formula $\phi \in \mathcal{F}$ leads to the modification of the possibility distribution (11) incurred by this formula. It is modified in such a way that the valuations not satisfying ϕ also get some non-zero possibility (for details see below; cf. (14)).

In the possibilistic logic the syntax of the classical propositional calculus is extended so as to make it possible to express the belief degree associated with a formula. So-called *weighted formulas* are introduced:

$$(\phi, \alpha) \tag{13}$$

where the weight $\alpha \in [0, 1]$ denotes the lower bound on the belief as to the truth of ϕ . It is worth to contrast the weighted formulas of the possibilistic logic with the evaluated formulas in the Rational Pavelka Logic, mentioned earlier. In the latter case the setting is that of a multivalued logic and the weights correspond to the truth values while in the former case the setting is that of the classical binary logic and the weights correspond to the belief degrees.

A weighted formula (13) induces the following possibility distribution on the set Ω :

$$\pi_{(\phi, \alpha)}(\omega) = \begin{cases} 1 & \text{for } \omega \in \Omega^\phi \\ 1 - \alpha & \text{for } \omega \notin \Omega^\phi \end{cases} \tag{14}$$

If a weighted formula (13) holds, i.e., represents an agent's knowledge, then this agent's belief in the truth of a formula $\psi \in \mathcal{F}$ is expressed by the pair of numbers (12), where Π_ϕ and N_ϕ are induced by the possibility distribution (14).

In general it is assumed that knowledge is expressed by a set of weighted formulas $P = (\phi_i, \alpha_i)$, referred to as the *belief base*. These formulas jointly induce the following possibility distribution on Ω :

$$\pi(\omega) = \min_i \pi_{\phi_i, \alpha_i}(\omega) \tag{15}$$

where π_{ϕ_i, α_i} is a possibility distribution induced by a formula (ϕ_i, α_i) , cf. (14).

A possibility distribution π is said to *satisfy* a formula (ψ, α) , which is denoted as $\pi \models (\psi, \alpha)$, if $N(\psi) \geq \alpha$, where N is the necessity measure related to the possibility distribution π . The logical consequence relation \models between the formulas of

possibilistic logic is defined as follows:

$$(\phi, \alpha) \models (\psi, \beta) \Leftrightarrow \forall \pi \quad \pi \models (\phi, \alpha) \Rightarrow \pi \models (\psi, \beta) \quad (16)$$

Thus, in particular:

$$\forall \alpha \geq \beta \quad (\psi, \alpha) \models (\psi, \beta) \quad (17)$$

Finally, some approaches in IR are based on a logic which may be called the genuine fuzzy logic in the sense of Zadeh. In this paper we will refer to it as *Zadeh's calculus of linguistic statements*. It is closely related to the concept of a linguistic variable¹³ and approximate reasoning.¹⁴ This formalism does not directly adopt the traditions of the classical logic. The concept of the formula is here replaced by the concept of a *linguistic statement*^{13,12}:

$$X \text{ IS } A \quad (18)$$

where X is a linguistic variable and A is a linguistic term which is declared in (18) to be the value of X . We will often identify A with a fuzzy set that represents its meaning. Such a fuzzy set A is defined in the universe U which is the domain of the *base variable* associated with the linguistic variable X . For example, in the linguistic statement “John is young” the age of John is treated as a linguistic variable AGE and “young” is the value assigned to it, which may be represented by the fuzzy set YOUNG defined, e.g., by the following membership function:

$$\mu_{YOUNG}(x) = \begin{cases} 1 & \text{for } x \leq 25 \\ \frac{35-x}{10} & \text{for } 25 < x \leq 35 \\ 0 & \text{for } x > 35 \end{cases}$$

defined in the domain $[0, 100]$ of the base variable associated with the linguistic variable AGE.

Thus, the linguistic statement (18) makes it possible to represent imprecise information about the value of a variable. If this information is additionally uncertain, then a qualified version of the statement (18), a so-called *certainty qualified statement* may be useful:

$$X \text{ IS } A \text{ is at least } \alpha\text{-certain} \quad (19)$$

what may be shorter denoted as

$$X \text{ IS } A, \alpha \quad (20)$$

The statement (19) may be identified with a simple statement $X \text{ IS } A'$ with $\mu'_{A'}$ defined as, e.g.,¹⁴:

$$X \text{ IS } A, \alpha \quad \mapsto \quad X \text{ IS } A' \quad (21)$$

$$\mu_{A'} = f(\alpha, \mu_A) \quad (22)$$

where function f may take different forms¹⁴, e.g.:

$$\mu_{A'}(x) = \max(\mu_A(x), 1 - \alpha) \quad (23)$$

It may be easily noticed that if an agent is completely certain as to the truth of (19), i.e., $\alpha = 1$, then due to (23) $A' = A$, i.e., $X \text{ IS } A, 1 \mapsto X \text{ IS } A$. Thus the representation (19) is a genuine generalization of (18). An interesting extension of (19) has been recently proposed¹⁵, where the number α may be replaced with a fuzzy number (linguistic term) expressing the certainty.

If the agent's knowledge is expressed by a statement (18), then he or she does not know which is an exact value of the associated base variable. Referring to our previous example: knowing that "John is young" does not tell which is the exact age of John. The age of John is *uncertain*. This uncertainty is quantified by the possibility distribution π_A on U generated by the assignment of A to X ; $\pi_A(x) \in [0, 1]$. It is assumed that

$$\pi_A(x) = \mu_A(x) \quad \forall x \in U. \quad (24)$$

$$\pi_A : U \longrightarrow [0, 1] \quad (25)$$

$$\pi_A(x) = \mu_A(x) \quad (26)$$

We will say that $X \text{ IS } A$ generates a possibility distribution π_A .

The total knowledge of an agent may be represented by a combination of statements (18), notably by a conjunction:

$$X_1 \text{ IS } A_1 \wedge \dots \wedge X_n \text{ IS } A_n \quad (27)$$

each $X_i \text{ IS } A_i$ generating a possibility distribution π_{A_i} and jointly generating a possibility distribution π on $U = U_1 \times \dots \times U_n$ such that $\forall x = (x_1, \dots, x_n) \in U$:

$$\pi(x) = \min(\pi_{A_1}(x_1), \dots, \pi_{A_n}(x_n)) \quad (28)$$

assuming the *non-interactiveness*¹² of the particular variables X_i .

In the calculus of linguistic statements the concept of valuation is not used directly. Instead it is considered what the truth of the statement $X \text{ IS } B$ is assuming the truth of $X \text{ IS } A$. The answer to such a question takes the form of a *fuzzy truth value*¹⁴, i.e., a fuzzy set defined in the interval $[0, 1]$. Often, for the sake of simplicity, a fuzzy truth value is replaced by a pair of the possibility and necessity measures ($\Pi_A(B), N_A(B)$):

$$\Pi_A(B) = \sup_{x \in U} \min(\pi_A(x), \mu_B(x)) \quad (29)$$

$$N_A(B) = \inf_{x \in U} \max(1 - \pi_A(x), \mu_B(x)) \quad (30)$$

related to the possibility distribution π_A generated by $X \text{ IS } A$; cf. (26).

4. Some Fuzzy Logic Based IR Models and Their Treatment of the Keyword Weights

In this paper we consider fuzzy logic based IR models as extensions to the classical Boolean model. In particular, we study how these models make it possible to

overcome a major drawback of the Boolean model – i.e., the binary interpretation of the importance of keywords in modelling both documents and user information needs (queries) – and still stay within the logical framework. Following the historical development of the fuzzy logic applications in IR we start with the case where non-binary weights are only used for the keywords in documents.

4.1. *Weighted keywords only in documents*

In the classic logical approach a document d may be identified with a (set of) valuation(s) ω ; cf. (2). Then, the assignments $\omega(s_i) = 1$ and $\omega(s_i) = 0$ mean that the keyword t_i is *important* and is *not important*, respectively, for the representation of the document. Hence by assuming that fuzzy logic is meant as a multivalued propositional logic (cf., p. 3; (4)) and the valuation is:

$$\omega : S \rightarrow [0, 1] \quad (31)$$

one immediately obtains the notion of a gradual importance of a keyword for the representation of a document. Thus now a document d is represented by a valuation ω^d and $\omega^d(s_i) \in [0, 1]$ denotes how important the keyword t_i is for this representation. The form of a query is the same as in the classical Boolean model. The *matching degree* of a query q and a document d is defined as the truth value of q under the valuation ω^d representing d .

Thus the multivalued logic provides a formal framework for the notions of a gradual importance and matching (relevance). How the weights are actually assigned to the keywords representing documents is a question which goes beyond the model as such. In general, a function F will be assumed:

$$F : D \times T \longrightarrow [0, 1] \quad (32)$$

In particular, one of the weighting schemes developed in the framework of the vector space model may be adopted.¹⁶ A popular choice for F might be a well-known $tf \times IDF$ scheme.

4.2. *Weighted keywords also in queries*

Thus to model the keywords weights in documents a simple fuzzy propositional logic is sufficient. In order to model the weighted keywords in queries a more sophisticated logical framework is needed. The Rational Pavelka Logic (RPL) (cf., p. 3; (5)) well serves the goal. The idea is to use the logical constants $\{\mathbf{a} \mid a \in [0, 1]\}$ (cf. p. 3) to express the weights. If a keyword t_i is to be assigned the weight $w_i^q \in [0, 1]$ in a query it is expressed via an evaluated formula (s_i, \mathbf{w}_i^q) . Thus queries with weighted keywords are still legitimate formulas of the underlying logic. Moreover the above mentioned evaluated formula may be expressed as the following “regular” formula:

$$\mathbf{w}_i^q \Rightarrow s_i \quad (33)$$

Then if in a document d the keyword t_i is assigned the weight w_i^d (i.e., this document is represented by the valuation ω^d such that $\omega^d(s_i) = w_i^d$) then the matching degree between the document d and query q is computed as $w_i^q \rightarrow w_i^d$, where “ \Rightarrow ” is an operator modeling the implication connective “ \Rightarrow ” (in the RPL it is originally the Lukasiewicz implication operator).

Now, there is an important question of what is the meaning of weights assigned to the keywords in a query. It turns out that for various operators “ \rightarrow ” modeling “ \Rightarrow ” in (33) one obtains some well-known *semantics* of these weights.² The choice of an implication operator is closely related (e.g., via the residuation) to the choice of an operator modelling the conjunction. In fact the RPL is an extension of the Lukasiewicz logic (cf. ⁸) with its specific conjunction operator and its related implication operator. A similar extension may be done for the Gödel and Product logics (cf. ⁸) (although with a resulting logic not possessing all the attractive properties of the RPL, but this goes beyond the scope of this paper). In what follows we will precisely specify the particular logic meant, wherever necessary.

In the following discussion we will assume the whole query has a form of a conjunction of weighted keywords:

$$(\mathbf{w}_1^q \Rightarrow s_1) \wedge \dots \wedge (\mathbf{w}_n^q \Rightarrow s_n) \quad (34)$$

For the Gödel logic:

$$x \rightarrow y = \begin{cases} 1 & \text{if } x \leq y \\ y & \text{otherwise} \end{cases} \quad (35)$$

for the Product logic (the Goguen implication):

$$x \rightarrow y = \min(y/x, 1) \quad (36)$$

and for the Lukasiewicz logic:

$$x \rightarrow y = \begin{cases} 1 & \text{if } x \leq y \\ 1 - x + y & \text{otherwise} \end{cases} \quad (37)$$

one obtains the *threshold semantics* of the importance weights. Namely, if the weight w_i^d of a keyword t_i in a document is higher or equal w_i^q , then the document d and the query q fully (to a degree = 1) match with respect to keyword t_i . If w_i^d does not reach *the threshold* set by w_i^q then the matching degree falls below 1, according to the respective implication operator.

For the Kleene-Dienes implication:

$$x \rightarrow y = \max(1 - x, y) \quad (38)$$

one obtains the *relative importance semantics*. Namely, a keyword with weight $w_i^q = 0$ in a query yields the matching degree equal 1 whatever is the corresponding weight w_i^d in a document. Thus it does not influence at all the overall matching degree of the whole query (34). On the other hand if $w_i^q = 1$ then the corresponding weight w_i^d in a document has to be high to contribute to the matching of the whole query

(34). Keywords with intermediate importance weights w_i^q contribute to the overall matching to some extent.

The threshold semantics seems to be less intuitive in the case when a low value for the keyword weight w_i^q is specified in a query. Why a user should come up with such a requirement? While addressing this problem, Herrera-Viedma¹⁷ proposed a modified version of this semantics. Namely, high query weights are treated as previously, as lower bounds on the weights of keywords in the documents, but low weights are treated as upper bounds on these weights. We propose another interpretation of this semantics in the framework of a model that we have developed recently and which is briefly discussed in p. 4.3.

Finally, the third semantics of *ideal weights*² requires the documents to have weights of particular keywords similar to the weights specified for these keywords in a query. This semantics does not fit the query representation schema (34). In Zadrozny and Kacprzyk¹⁸ we proposed to use, instead of (34), the following form of the query, if such a semantics is called for:

$$(\mathbf{w}_1^q \Leftrightarrow s_1) \wedge \dots \wedge (\mathbf{w}_n^q \Leftrightarrow s_n) \quad (39)$$

where the equivalence connective “ \Leftrightarrow ” is to be modelled by the following operator “ \leftrightarrow ”:

$$(x \leftrightarrow y) = \min(x \rightarrow y, y \rightarrow x) \quad (40)$$

and “ \rightarrow ” is the Goguen implication operator given by (36).

It should be noted that other forms of the query, e.g., where disjunction replaces conjunction, require a different from (34) formulation of the query. In particular for a query being a disjunction of the weighted keywords the following formal representation may be assumed:

$$(\neg \mathbf{w}_1^q \Rightarrow_c s_1) \vee (\neg \mathbf{w}_2^q \Rightarrow_c s_2) \vee \dots \vee (\neg \mathbf{w}_K^q \Rightarrow_c s_K) \quad (41)$$

where “ \Rightarrow_c ” is a *coimplication* connective. This connective may be treated as a derived one whose semantics in the context of the lattice (5) may be provided by an operator \rightarrow_c defined as follows¹⁹:

$$x \rightarrow_c y = \neg(\neg x \rightarrow \neg y)$$

Thus, for the Kleene-Dienes, Gödel and Goguen implication operators the corresponding coimplication operators are defined, respectively:

$$\begin{aligned} x \rightarrow_c y &= \min(1 - x, y) \\ x \rightarrow_c y &= \begin{cases} 0 & \text{if } x \geq y \\ y & \text{otherwise} \end{cases} \\ x \rightarrow_c y &= \begin{cases} 0 & \text{if } x = 1 \\ \max\{0, \frac{y-x}{1-x}\} & \text{otherwise} \end{cases} \end{aligned}$$

For other implication operators, which are widely employed, the derivation of their corresponding coimplication operators may proceed in a similar way.

Thus, in the framework of the RPL or its variants the most popular semantics of keywords weights are easily expressible via more or less complex logical formulas. It should be stressed that the use of the implication connective to model the matching between a query and a document has been thoroughly studied in the past (cf., e.g., papers by Bordogna et al.^{20,21}). However, these studies were focused on the “semantic”, quantitative aspect of the matching. Here we aim at providing a comprehensive treatment of the issue, addressing its both semantic and syntactic aspects in a strict logical framework.

4.3. *Weights specified imprecisely and without certainty*

The approaches discussed so far assume the weights to be expressed using numbers. In practice it may be inconvenient. Thus it has been proposed to use *linguistic terms*, modelled with fuzzy sets in interval $[0, 1]$, to express weights of the keywords in queries^{22,23,17,24,25}. Recently we proposed a model in which also the weights of keywords in *documents* may be specified this way, cf. Nowacka, Kacprzyk, Zadrozny^{26,27}. This model is based on fuzzy logic meant in another sense than in the previous sections. Namely, the Zadeh calculus of the linguistic statements is employed (cf. Section 3). The characteristic features of this model are as follows:

- documents and queries are represented in an uniform way using linguistic terms to express the keyword weights,
- both imprecision and uncertainty concerning the specification of keyword weights may be accounted for,
- the matching degree between a document and a query is defined using possibility theory.

In this model the importance of each keyword t_i in a document/query is assumed to be a *linguistic variable* X_i ¹³. The domain of the base variable (cf. (18) and comments thereunder) associated with this linguistic variable is the interval $[0, 1]$, i.e. the universe of the importance degrees as assumed in the previously discussed fuzzy models. The weight of a keyword is expressed by a *linguistic term* A_i such as *very important* or *important to a degree around 0.6*, etc. The importance of a keyword t_i is expressed by a statement of Zadeh’s calculus of linguistic statements^{13,12}:

$$X_i \text{ IS } A_i \tag{42}$$

With such a statement (42) a possibility distribution π_{A_i} (cf. (24)) is associated which models the uncertainty as to which of the values from the interval $[0, 1]$ expresses the actual importance of given keyword in a document.

In order to additionally grasp the uncertainty as to the actual importance of a keyword a certainty qualified statement (19) may be used expressing the minimal degree of certainty $\alpha \in [0, 1]$. This in turn may be translated into a simple statement (42) using the rules expressed by (21)–(22), in a particular case by (23).

The whole document/query is then represented by a combination of statements (18), notably by a conjunction:

$$X_1 \text{ IS } A_1 \wedge \dots \wedge X_n \text{ IS } A_n \quad (43)$$

each $X_i \text{ IS } A_i$ generating a possibility distribution π_{A_i} and jointly generating a possibility distribution (44) π on $U = [0, 1]^n$ such that $\forall x = (x_1, \dots, x_n) \in U$ (this is the same possibility distribution as in (28) - but there discussed in a more general context and recalled here for the sake of an easier “navigation” through the paper):

$$\pi(x) = \min(\pi_{A_1}(x_1), \dots, \pi_{A_n}(x_n)) \quad (44)$$

In (44) the non-interactiveness of the keywords importance weights is assumed. This is a simplifying assumption which is a counterpart of the probabilistic independence assumption often adopted in the IR literature. Thus $X = (X_1, \dots, X_n)$ is a multidimensional linguistic variable representing the importances of all keywords in documents and queries. However, for clarity we will further on focus on the representation of a document/query given by a single statement (42). A more comprehensive treatment of the representation assumed in our model may be found in our papers ^{26,27}.

The matching between a query and a document is determined as follows. The statement $X \text{ IS } A$ representing a document d is meant to generate a possibility distribution $\pi_A : [0, 1] \rightarrow [0, 1]$, $\pi_A(x) = \mu_A(x)$ and the matching degree of this document against a query q represented by $X \text{ IS } B$ is computed as a pair of values $(\Pi_A(B), N_A(B))$:

$$\Pi_A(B) = \sup_{x \in U} \min(\pi_A(x), \mu_B(x)) \quad (45)$$

$$N_A(B) = \inf_{x \in U} \max(1 - \pi_A(x), \mu_B(x)) \quad (46)$$

Now we will show how some *templates* ²⁶ of the linguistic values A and B in (42), (45) and (46) correspond to various semantics of keywords weights discussed earlier (cf. Section 4.2). Each template has one parameter, a number from the interval $[0, 1]$. From now on these parameters will be denoted with the same symbols as the document or the query, i.e., d_i and q_i for linguistic terms A_i and B_i , respectively, representing importance of the keyword t_i in a document and in a query.

The weights of all keywords in all documents are expressed using the same template with varying value of the parameter and similarly for the query. Different templates may be used for the documents and queries, thus some variants may be distinguished by selecting two templates.

Variante I Linguistic terms A_i (in documents) correspond to the expression *important to a degree around d_i* and are represented by triangular membership functions parametrized with $d_i \in [0, 1]$:

$$\mu_{A_i}(x) = 1 - |d_i - x| \quad \forall x \in U \quad (47)$$

Thus one importance degree (d_i) is fully possible while for the other degrees, the further they are from d_i the less they are possible as actually expressing the importance of the keyword under consideration.

Linguistic terms B_i (in queries) correspond to the expression *important with certainty at least q_i* and are represented by the following membership function:

$$\mu_{B_i}(x) = \begin{cases} 1 - q_i & \text{for } x \leq 1 - q_i \\ x & \text{for } x > 1 - q_i \end{cases} \quad \forall x \in U \quad (48)$$

The underlying membership function $\mu_{A_i}(x) = x$ refers to the multivalued (non-binary) concept “important”: the higher the degree of importance the more compatible it is with this term. It is combined with the expression of some uncertainty as to the actual importance of the keyword; cf. (19). The rationale for the use of this template is the following. It is assumed that in a query only the important keywords are used. However the user may have some doubts if a given keyword is really important. His or her limited certainty is expressed by a number q_i , corresponding to α in the formula (19).

The matching is determined by the formulas (45)–(46), which for the templates used in this variant take the following form:

$$\Pi_{A_i}(B_i) = \begin{cases} 1 - q_i & \text{for } q_i \leq \frac{1-d_i}{2} \\ \frac{1+d_i}{2} & \text{for } q_i > \frac{1-d_i}{2} \end{cases} \quad (49)$$

$$N_{A_i}(B_i) = \begin{cases} 1 - q_i & \text{for } q_i \leq 1 - \frac{d_i}{2} \\ \frac{d_i}{2} & \text{for } q_i > 1 - \frac{d_i}{2} \end{cases} \quad (50)$$

Let a query be in the form of a conjunction of n statements X_i IS B_i , where the membership functions of B_i 's are given by (48). When q_i tends to 0, then the membership function μ_{B_i} approaches $\mu_{B_i}(x) = 1, \forall x \in U = [0, 1]$. Then the values of the possibility and necessity measures (49)–(50) both tend to 1, *independently* of the keyword weight in the document. Thus the keyword t_i has a lower and lower influence on the matching degree computation for this query and any document, due to the minimum operator used in (28). The opposite happens for q_i approaching 1 – then the high value of d_i is required to obtain the match. Hence in variant I we recover the *semantics of the relative importance*; cf. Section 4.2. Even if the minimum operator in (44) is replaced with another one (which has been suggested by our computational experiments²⁷) still this semantics may be preserved. For example, this will be the case of the arithmetic mean replacing the minimum operator. It should be noted that the absolute value of the matching degree is not that important – what really counts is the ordering of documents implied by the matching degrees. Thus, if for q_i close to 0 all documents get a high matching degree along the attribute t_i this does not influence the implied ordering.

Variante II In this variant the templates for documents and queries are identical. Their membership functions are as given by (47) in Variant I. The formulas for the

matching degree (45)–(46) take for these templates the following form:

$$\Pi_{A_i}(B_i) = 1 - \frac{|d_i - q_i|}{2} \quad (51)$$

$$N_{A_i}(B_i) = \begin{cases} \frac{1-q_i+d_i}{2} & \text{for } (q_i \geq 1-d_i) \ \& \ (q_i \geq \frac{d_i+1}{3}) \\ q_i & \text{for } (q_i \geq 1-d_i) \ \& \ (q_i \leq \frac{d_i+1}{3}) \\ 1 - q_i & \text{for } (q_i \leq 1-d_i) \ \& \ (q_i \geq \frac{d_i+1}{3}) \\ \frac{1-d_i+q_i}{2} & \text{for } (q_i \leq 1-d_i) \ \& \ (q_i \leq \frac{d_i+1}{3}) \end{cases}$$

The obtained values of the necessity measure may be seen as rather counter-intuitive for some ranges of values d and q . For example, for $d = q = 1$ one obtains a lower degree of necessity than for $d = 1$ and $q = 0.9$. Hence in this variant it may be reasonable to take into account the value of the possibility measure (51) only.

If the possibility measure (51) is used alone, then the computed matching degree is the higher the closer the values of q and d are. Thus in Variant II we recover the *semantics of ideal weights*.

Variant III This variant is inspired by the possibilistic logic. As briefly discussed in Section 3, this is basically a binary logic but making possible the expression of a limited belief (certainty) in the truth of a formula. Thus in Variant III we assume that the keywords may only be either important or not important. Hence the sets A_i and B_i are crisp single-element sets $\{0\}$ or $\{1\}$. When the limited certainty α is taken into account they are transformed using equation (23) into, in the general case, fuzzy sets in space $\{0, 1\}$.

Thus it is assumed that to represent both documents and queries only important keywords t_i are selected. However it may be not completely certain if they are really important. In other words it is *completely possible* (with the possibility degree equal 1) that a keyword t_i is important but it is still possible to some extent (the possibility degree greater than 0) that it is not *important*.

Hence the template used in Variant III may be described using the following membership function:

$$\mu_{A_i}(x) = \begin{cases} 1 - d_i & \text{for } x = 0 \\ 1 & \text{for } x = 1 \end{cases} \quad \forall x \in U \quad (52)$$

Here d_i is a parameter corresponding to α in formula (19) and indicating a lower bound of the certainty as to the importance of given keyword t_i .

In Variant III the importance of keywords in queries is represented by exactly the same template as for the documents. Thus we have:

$$\mu_{B_i}(x) = \begin{cases} 1 - q_i & \text{for } x = 0 \\ 1 & \text{for } x = 1 \end{cases} \quad \forall x \in U \quad (53)$$

So q_i plays here the same role as d_i in (52).

The formulas for the matching degree (29)–(30) take for these templates the following form:

$$\Pi_{A_i}(B_i) = 1 \quad (54)$$

$$N_{A_i}(B_i) = \begin{cases} 1 - q_i & \text{for } q_i \leq 1 - d_i \\ d_i & \text{for } q_i > 1 - d_i \end{cases} \quad (55)$$

Thus, in fact only the necessity measure is helpful in deciding on the relevance of a document. Moreover the formula (55) may be rewritten as $N_{A_i}(B_i) = \max(1 - q_i, d_i)$ what directly refers to (38) and shows that here we have again the *relative weights semantics*.

However let us look at this variant from the perspective of the propositional possibilistic logic more closely. Then both the documents and queries are represented using the *weighted formulas* of this logic: (s_i, α) , where $s_i \in S$ is a usual propositional variable and α expresses a limited belief in its truth. Then the logical dependency $d \Rightarrow q$ (cf. (3)) may be meant here as the semantic consequence operation $d \models q$. In the possibilistic logic (cf. (16))

$$(s, \alpha) \models (s, \beta)$$

if and only if $\alpha \geq \beta$ (cf. (16)–(17)). Here α and β correspond to d_i and q_i , respectively, in (55). Thus, if we adopt this interpretation we obtain the *threshold semantics* of the keywords weights. At the same time we get some rationale for this semantics. Namely, if the user finds a keyword t_i important for the query but is not totally sure as to this importance, then a document will fit this query with respect to t_i unless certainty in this keyword's importance for document representation is even lower.

5. Concluding remarks

We have shown how fuzzy logic may be applied for information retrieval modelling, more in the spirit of the formal logical models. In particular we have offered some formal means to represent various known semantics of the keywords importance. An important contribution of this paper seems to be that we have shown that a combination of fuzzy logic in the narrow and broad sense, and of a possibilistic logic, may offer new vistas and synergistic, effective and efficient tools for the definition of semantics of the weights of keywords which is crucial, still unsolved a problem in IR.

References

1. M. Lalmas, "Logical models in information retrieval: Introduction and overview", *Information Processing & Management* **34** (1998) 19–33.
2. G. Bordogna, P. Carrara and G. Pasi, "Query term weights as constraints in fuzzy information retrieval", *Information Processing & Management* **27** (1991) 15–26.

3. R. Baeza-Yates and B. Ribeiro-Neto, *Modern information retrieval* (ACM Press and Addison Wesley, 1999).
4. F. Crestani and M. Lalmas, “Logic and uncertainty in information retrieval”, in *Lectures on information retrieval* (Springer-Verlag New York, Inc., New York, NY, USA, 2001), pp. 179–206.
5. C. J. van Rijsbergen, “A new theoretical framework for information retrieval”, in *Proc. of ACM SIGIR Conference on Research and Development in Information Retrieval*, ed. Rabitti F. (Pisa, Italy, 1986), pp. 194–200.
6. C. J. van Rijsbergen, “A non-classical logic for information retrieval”, *The Computer Journal* **29** (1986) 481–485.
7. F. Sebastiani, “A note on logic and information retrieval”, in *MIRO’95 Proc. of the Final Workshop on Multimedia Information Retrieval* (Springer, Glasgow, Scotland, 1995).
8. P. Hajek, “On the metamathematics of fuzzy logic”, in *Discovering the World with Fuzzy Logic*, eds. V. Novak and I. Perfilieva (Physica-Verlag, Heidelberg New York, 2000), pp. 155–174.
9. V. Novak, I. Perfilieva and J. Močkoř, *Mathematical Principles of Fuzzy Logic* (Kluwer, Boston, 1999).
10. I. Perfilieva and V. Novak, “Fuzzy logic on the basis of classical logic”, in *Selected Topics on Information Technology*, eds. J. Kacprzyk, M. Krawczak and S. Zadrozny (EXIT, 2002), pp. 83–130.
11. D. Dubois, J. Lang and H. Prade, “Possibilistic logic”, in *Handbook of Logic in Artificial Intelligence and Logic Programming*, ed. Gabbay D.M. et al. (Oxford University Press, Oxford, UK, 1994), volume 3, pp. 439–513.
12. L. Zadeh, “Fuzzy sets as a basis for a theory of possibility”, *Fuzzy Sets and Systems* **1** (1978) 3–28.
13. L. Zadeh, “The concept of a linguistic variable and its application to approximate reasoning. Part I-III.”, *Information Sciences* **8,8,9** (1975) 199–249,301–357,43–80.
14. D. Dubois and H. Prade, “Fuzzy sets in approximate reasoning, part 1: Inference with possibility distributions”, *Fuzzy Sets and Systems* **40** (1991) 143–202.
15. A. González, N. Marín, O. Pons and M. Vila, “Fuzzy certainty on fuzzy values”, *Control and Cybernetics to appear*.
16. G. Salton and C. Buckley, “Term-weighting approaches in automatic text retrieval”, *Information Processing and Management* **24** (1988) 513–523.
17. E. Herrera-Viedma, “Modeling the retrieval process of an information retrieval system using an ordinal fuzzy linguistic approach”, *JASIST* **52** (2001) 460–475.
18. S. Zadrozny and J. Kacprzyk, “An extended fuzzy Boolean model of information retrieval revisited”, in *Proc. of FUZZ-IEEE 2005* (IEEE, Reno, NV, USA, May 22–25, 2005), pp. 1020–1025.
19. J. Fodor and M. Roubens, *Fuzzy Preference Modelling and Multicriteria Decision Support*, Sereis D: System Theory, Knowledge Engineering and Problem Solving (Kluwer academic Publishers, 1994).
20. G. Bordogna, P. Bosc and G. Pasi, “Fuzzy inclusion in database and information retrieval query interpretation”, in *Proceedings of the 1996 ACM symposium on Applied Computing* (ACM, 1996), p. 547551.
21. G. Bordogna, P. Bosc and G. Pasi, “Extended Boolean information retrieval in terms of fuzzy inclusion”, in *Knowledge Management in Fuzzy Databases*, eds. O. Pons, M. Vila and J. Kacprzyk (Physica Verlag, Heidelberg New York, 2000), pp. 234–246.
22. G. Bordogna, P. Carrara and G. Pasi, “Fuzzy approaches to extend Boolean information retrieval”, in *Fuzziness in Database Management Systems*, eds. P. Bosc and

18 *S. Zadrożny, J. Kacprzyk*

- J. Kacprzyk (Physica Verlag, Heidelberg, 1995), pp. 231–274.
23. D. Kraft, G. Bordogna and G. Pasi, “An extended fuzzy linguistic approach to generalize Boolean information retrieval”, *Journal of Information Sciences* **2** (1994) 119–134.
 24. E. Herrera-Viedma and A. López-Herrera, “A model of information retrieval system with unbalanced fuzzy linguistic information”, *International Journal of Intelligent Systems* **22** (2007) 1197–1214.
 25. A. López-Herrera, E. Herrera-Viedma and F. Herrera, “Applying multi-objective evolutionary algorithms to the automatic learning of extended boolean queries in fuzzy ordinal linguistic information retrieval systems”, *Fuzzy Sets and Systems* **to appear**.
 26. K. Nowacka, S. Zadrożny and J. Kacprzyk, “A new fuzzy logic based information retrieval model”, in *12th International Conference Information Processing and Management of Uncertainty in Knowledge-based Systems (IPMU'08)* (Malaga, Spain, 2008).
 27. K. Nowacka, S. Zadrożny and J. Kacprzyk, “An experimental comparison of various aggregation operators in a fuzzy information retrieval model”, in *2008 Annual Meeting of the North American Fuzzy Information Processing Society (NAFIPS 2008)* (New York, USA, 2008).