Jason Agents for Knowledge-aware Information Retrieval Filters

Dima El Zein[a] and Célia da Costa Pereira[b]
Laboratoire i3s, Université Côte d'Azur, CNRS, UMR 7271, Sophia Antipolis, France

Keywords: Agent Based Framework, Knowledge Aware Filter, User Knowledge, Information Retrieval, BDI Agents.

Abstract: This paper proposes a novel use of Belief-Desire-Intention agents in Information Retrieval. We present a cognitive agent that builds its beliefs about the user's knowledge during his/her interaction with the search system. The agent reasons about those beliefs and derives new ones using contextual rules. Those beliefs serve to personalise the search results accordingly. The framework is developed using an extended version of the Jason agent programming language; the choice of Jason's extension to model the agents is justified by some of its advantageous features, in particular, the possibility to represent gradual beliefs. A running example will illustrate the presented work and highlight its added value.

1 INTRODUCTION, MOTIVATION AND RELATED WORK

It is increasingly common for search and recommendation systems to personalise the items proposed according to the user's preferences, location, profile, etc. However, most of these systems build the user's profile based on his/her search history and do not consider the evolution of the user's information needs from a "cognitive point of view" (Culpepper et al., 2018). For example, some existing contributions in personalisation applied content-based techniques by using the content of the documents read by the user to construct the user's profile (Ricci et al., 2011; Garcin et al., 2012). The related user profiles are often static or not frequently updated, hence they cannot help represent the user's knowledge, which is constantly evolving. The constant evolution of the user's knowledge is an important aspect to be considered when proposing information that is supposed to be novel and/or helpful for the user to achieve a goal. In our opinion, this gap should be essentially filled: search results must be adapted to the users' beliefs, the knowledge they acquire, or even the goals they want to achieve. The difficulty of extracting, representing, and measuring the cognitive aspects may explain this gap. Indeed, to the best of our knowledge, there are only a few implemented work in Information Retrieval (IR) systems that considers "understanding" the user's mental attitudes (belief and knowledge changes, goals, ...) and personalising the results accordingly. We believe that to be promising, an approach should consider users as cognitive agents (da Costa Móra et al., 1998; Rao and Georgeff, 2001) with their own beliefs and knowledge of the world.

The first contributions using an agent-based architecture in IR have been proposed twenty years ago; the main goal was to track the user's activities to personalise Web search (Guttman and Maes, 1998; Bakos, 1997). Then, in order to better understand the user's behaviour during the search, some user-related characteristics, such as location and type of device, were considered (Carrillo-Ramos et al., 2005; Kurumatani, 2004). In (Yu et al., 2021), the authors explored the correlation between the content of a document read and the search behaviour from the one hand, and a user's knowledge state and knowledge gain from the other hand. The results showed that the knowledge gain can be predicted from the users' search behaviour and from the content features of the documents they read (e.g., number of money words, number of religion words, number of words in each page, etc.).

A theoretical proposal for an IR framework considering the user's knowledge to personalise the search result was recently proposed in (El Zein and da Costa Pereira, 2020a). The scope of the framework was textual document retrieval where the user's knowledge was assumed to be the informa-

[a] https://orcid.org/0000-0003-4156-1237
[b] https://orcid.org/0000-0001-6278-7740

El Zein, D. and Pereira, C.
Jason Agents for Knowledge-aware Information Retrieval Filters.
DOI: 10.5220/0010884900003116
In *Proceedings of the 14th International Conference on Agents and Artificial Intelligence (ICAART 2022) - Volume 2*, pages 466-476
ISBN: 978-989-758-547-0; ISSN: 2184-433X

tion contained in the document(s) he/she was reading. The framework considered an agent, designed in a Belief-Desire-Intention (BDI) architecture (Rao and Georgeff, 2001), and was made "aware" of the user's knowledge. The agent's beliefs represent the user's knowledge that is extracted from the documents the user has read. Beliefs are considered to be gradual and are associated to some degrees that define the extent to which a piece of knowledge is "entrenched". These degrees represent the preferences between beliefs, and are used in case of divergence or contradiction between new and old beliefs. In particular, they help deciding which belief to maintain and which to discard in case of contradiction. The agent also has some contextual rules (or knowledge rules) that are used to derive new beliefs. The main task is to "filter" the search results returned by an IR system to fit the user's knowledge.

In (El Zein and da Costa Pereira, 2020b) the authors extended the Jason platform (Bordini et al., 2007) in order to account for the concept of graded beliefs, implement belief reasoning and belief changing capabilities. The resulting approach allows agents to reason about the degree of certainty of beliefs, track the dependency between them and revise the belief set accordingly.

In this paper, we propose an Information Filter agent which puts together the advantages of the theoretical framework proposed in (El Zein and da Costa Pereira, 2020a) with the extended Jason platform proposed in (El Zein and da Costa Pereira, 2020b). We will develop and describe the implementation of such a framework, and discuss the advantages and challenges associated with it.

For example, if the user reads a document about *human evolution*, the agent will be aware of the user's knowledge of that subject. Then, suppose the user submits the query "Charles Darwin", the search system returns a list of documents on several topics about *Darwin*, such as his bibliography, his theory of natural selection and social Darwinism. Since the agent believes that its user already has knowledge about natural selection, it should decide which, among the documents relevant to the user's query, to return and which not to. For example, if the agent's purpose is to return new information, then documents dealing with the subject of *natural selection* will be discarded.

The language required to develop our filtering agent must allow the following: (1) representing the user's knowledge in the form of agent beliefs (2) associating beliefs to a degree of entrenchment (3) revising the beliefs in case of contradiction (4) updating the degree of a belief if it already exists (5) deriving new beliefs by reasoning with the knowledge rules. All the mentioned conditions are satisfied in the extended version of Jason (El Zein and da Costa Pereira, 2020b).

The paper is organised as follows. Section 2 discusses the preliminaries to understand the features of the framework. Section 3 describes Jason, its current limitations, and the features of its extended version. Section 4 discusses the cognitive IR framework. Section 5 describes the proposed implemented prototype as a proof-of-concept in the news articles (BBC) domain and, finally, some conclusions and perspectives of future work are presented in Section 6.

2 PRELIMINARIES

2.1 Rule-based Agents

A Rule-based agent (Jensen and Villadsen, 2015) has a belief base consisting of facts (ground literals) and rules (Horn clauses). The facts can originate from communication with other agents, observations of the environment, downloaded information from web sources or other ressources. Facts might change over time as a result of the inference process or of the addition and deletion of other facts from the agent's belief base. A literal α is a predicate symbol that is possibly preceded by a negation symbol $\neg$. We consider a finite set R of rules, which have of the form $\alpha_1 \& \alpha_2 \ldots \& \alpha_n \rightarrow \beta$ where $\alpha_1, \alpha_2, \ldots, \alpha_n (n \geq 1)$ *and* β are literals. β is called the derived belief, and each α_i is a premise of the rule. The & represents the logical *and* operator. We consider the agent's beliefs when the agent's rules have run to quiescence, i.e., after all the agent's rules have been applied to all the literals in the agent's memory. Note that this set is finite if the original set of rules and ground literals is finite.

2.2 Belief Revision and Partial Entrenchment Ranking

Belief revision is, by definition, the process of modifying the belief base to maintain its consistency whenever new information becomes available. The AGM belief revision theory (Alchourrón et al., 1985) defines postulates that a rational agent should satisfy when performing belief revision. In such a theory, a belief base is closed under logical consequence. We consider a belief base K and a new piece of information α. K is inconsistent, when both α and $\neg\alpha$ are in $Cn(K)$, or $Cn(K) = \perp$, or both α and $\neg\alpha$ are logical consequences of K. Three operators are considered: *Revision* $K * \alpha$: adds a belief α as long as it does not cause a contradiction in K. If the addition will cause inconsistencies in K, the revision operation starts by minimal changes in K to make it consistent with α, then adds α. *Expansion* $K + \alpha$: adds a new belief α that does not contradict with the existing beliefs. *Contraction* $K \div \alpha$: removes a belief α and all other beliefs that logically imply/entail it. The sentences in a belief set may not be considered equally important as it is assumed in the AGM postulates: belief is gradual and an agent might have beliefs more entrenched/accepted than others. Williams (Williams, 1995) have proposed a quantitative approach for the AGM framework, by developing finite partial entrenchment rankings to represent epistemic entrenchment – a piece of information is labelled by a degree of confidence denoting how strongly we believe it. The epistemic entrenchment (Gärdenfors and Makinson, 1988) captures the notions of significance, firmness, or defeasibility of beliefs.

Intuitively, epistemic entrenchment relations induce preference orderings of beliefs according to their importance in the face of change. If inconsistency arises during belief revision, the least significant beliefs (i.e., beliefs with the lowest entrenchment degree) are given up until consistency is restored. The belief revision operator(s) must then take into consideration the degree i of the belief to be added and decide whether to add it or not. We discuss later in Section 5.1 the belief revision algorithm $K * (\alpha, i)$ we followed.

2.3 Alechina's Belief Revision and Tracking

Alechina *et al.* (Alechina et al., 2005) proposed belief revision and contraction algorithms for resource-bounded agents. They consider a finite state and a finite program having a fixed number of rules used to derive new beliefs from the agent's existing beliefs. The approach associated a preference order (similar to Williams' approach (Williams, 1995)) for each belief and tracked dependencies between them.

For every fired rule instance, a Justification J will record: (i) the derived belief and (ii) a *support list*, s, which contains the premises of the rules. The dependency information of a belief had the form of two lists: *dependencies* and *justifications*. A *dependencies list* records the justifications of a belief, and a *justifications list* contains all the Justifications where the belief is a member of support. The approach represents the agent's belief base as a directed graph with two types of nodes: *Beliefs* and *Justifications*. A Justification has one outgoing edge to the belief it is a justification for, and an incoming edge from each belief in its support list.

Preference on Beliefs and Quality of justifications As beliefs are associated with preferences, justifications are associated with qualities. A quality of a justification is represented by non-negative integers in the range $[0, \ldots, m]$, where m is the maximum size of working memory. The lower the value, the least the quality.

Definition 1. *The preference value of a belief* α*,* $p(\alpha)$*, is equal to that of its highest quality justification.*

$$p(\alpha) = \max\{qual(J_0), \ldots, qual(J_n)\} \tag{1}$$

Definition 2. *The quality of justification* J*,* $qual(J)$*, is equal to the preference of the least preferred belief in its support list.*

$$qual(J) = \min\{p(\alpha) : \alpha \in \text{support of } J\} \tag{2}$$

Independent beliefs have at least one justification with an empty support list (non-inferential justification). They are usually those in the initial belief base or those perceived from the environment. It was assumed that non-inferential justification is associated with an *a priori* quality.

3 JASON: PROPERTIES, LIMITATIONS & EXTENSION

This section presents an overview of the Jason language's architecture (Bordini et al., 2007) and its features (v.2.4) (Jas, 2021). We also discuss the features and the motivation of the extended version proposed in (El Zein and da Costa Pereira, 2020b) that we will use later in our framework.

3.1 Architecture

A Jason agent, similarly to other agents modeled in BDI, is defined by sets of *beliefs*, *plans*, and *goals or intentions*. Jason's beliefs are represented by predicates. Their existence in the agent's belief base means the agent currently believes that to be true. The $\sim$ operator refers to the negation, explicitly representing that the agent believes a literal to be false. Annotations distinguish the Jason syntax: an annotation is a list of terms placed after a belief, enclosed in square brackets, revealing details about it.

A plan is composed of three parts: the *triggering event*, the *context*, and the *body*. It is expressed as follows:

$$+triggering\ event : context \leftarrow body. \tag{3}$$

Table 1: Comparison between Jason and its extension's features.

Feature	Original	Extended
Beliefs		
Dependencies	Not tracked	Tracked
Inconsistency	Accepted	Not accepted
Graded	No	Yes (degOfCert)
Preference[1]	No preference	High $\succ$ Low New $\succ$ Old
Plans		
Knowledge-rule with n conditions	n plans	1 plan
Order of conditions	Dep—endent	Independent (with +tei)
Execution of applicable plans with same triggering event	One plan only	All plans

triggering event represents one condition that might initiate the plan's execution; it can be the addition (+) or deletion (-) of a belief or a goal. *context* is a conjunction of literals that need to be satisfied to make the plan applicable – and possibly executed. A plan is applicable if: (i) first, its triggering event occurred, and (ii) its context (one or several conditions) is a logical consequence of the agent's current beliefs. The *body* is a sequence of actions or *goals* to be achieved upon the plan execution. Together the triggering event and the context constitute the plan's *head*.

3.2 Extension of Jason: Graded Belief Revision

The subject Jason extension (El Zein and da Costa Pereira, 2020b) models knowledge-rule agents that are capable of reasoning with uncertain beliefs, tracking the dependency between beliefs as done in (Alechina et al., 2005) and explained in Section 2.3 and revising the belief set accordingly. In the following, we highlight the limitations of the original version and how the extension overcame them. Table 1 compares the features of the two versions.

3.2.1 Representation and Execution of Knowledge Rules

To represent a rule having n conditions in the form of $\alpha_1 \& \alpha_2 \ldots \& \alpha_n \rightarrow \beta$ using Jason plans, the premises of rules are supposed to be in the plan's head. That means that one of the conditions of the premises must be the triggering event and the others in the context (for example, $+\alpha_1 : \alpha_2 \ \& \ldots \alpha_n \rightarrow \beta$). However, the plan execution in the original Jason version is reliant on the occurrence of the triggering event: a plan is executed only if the context conditions were satisfied before the triggering event takes place. The order of the triggering event and the context conditions matters for the execution of a plan. If the condition a_1 was satisfied before the others, the plan will not be executed. One alternative could be to write n plans. The extended version of Jason allows the expression of knowledge-rules by the so-called *Trigger-Independent plans*. Those plans will be executed whenever the combination of several conditions is satisfied, no matter which condition was satisfied first. In other terms, they do not wait for one specific trigger condition to occur to execute the plan. The syntax of *Trigger-Independent plans* should have the reserved word "+*tei*" that stands for *trigger event independent* in the trigger part and all the other conditions in the context. The plan's new syntax to represent knowledge rules is proposed:

$$+tei : context' \leftarrow body. \quad (4)$$

$context'$ has all the premises α_1 & $\alpha_2 \& \ldots \& \alpha_n$ and the *body* contains β the *derived belief*.

Using the original Jason in the case where two or more plans had the same trigger and all had a satisfying context field, would return only one plan for execution. The returned plan would be by default the first plan according to the order in which plans were written in the code. Contrarily, using the extension in the same case would return/execute all the plans having satisfying conditions.

3.2.2 Degree of Certainty

The notion of "believing" in Jason is Boolean: An agent either believes something is true or false or is ignorant about it. The extension allowed the representation of gradual beliefs by expressing "degOfCert(X)" in the annotation part of a belief - X represents the belief certainty defined as follows: We define the certainty of a belief α as representing the degree to which the agent believes the belief is true.

The degree of certainty associated with *initial beliefs*, *beliefs communicated* by other agents and *beliefs perceived* by the agent must be explicitly defined by the source. As for *derived beliefs* their related degree will be discussed in 3.2.3.

3.2.3 Deriving and Tracking Beliefs

Derived beliefs are dependent on the premises that derived them; therefore to calculate their related de-

[1] $\succ$ refers to "more preferred".

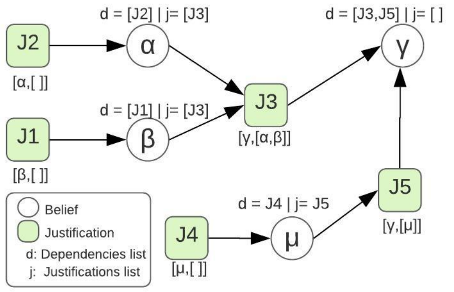

Figure 1: Graph over the beliefs and justifications.

gOfCert, the dependency between beliefs must be tracked. The extension tracked the beliefs following the approach discussed in 2.3: a justification is represented by a derived belief, a support list, and a quality; a belief is represented by a dependencies list, a justifications list and a degree of certainty. Whenever a knowledge-rule, named trigger-independent plan, is fired and a new belief is added, a justification node is created. This node links the rule's premises with the derived belief. The degree of a derived belief is automatically calculated by the interpreter using Equation 1. When any of the beliefs is contracted, the related justifications are removed as well. Justifications with an empty support lists are created upon the addition of initial, communicated, and perceived beliefs. Unlike in (Alechina et al., 2005), no *a priori* qualities are assigned for the justification of independent beliefs, as the degrees are explicitly stated.

Example 3.1. *Figure 1 illustrates the belief tracking, considering four beliefs* α, β, γ, *and* μ, *and a rule* α *&* $\beta \rightarrow \gamma$. *The rule means that if the agent believes in* α *and* β, *it believes in* γ. *For example, Justification* J_3 *is denoted as* (γ,[α, β]); γ *is the derived belief and* [α, β] *is the support list.* J_3 *is in the dependencies list of* γ *and in the justifications list both* α *and* β. *If* γ *were also derived from* μ, *i.e.* $\mu \rightarrow \gamma$, *then its dependencies list would also include another justification* J_5 *denoted as* (γ,[μ]). *If the belief* α *was the result of an observation, its dependencies list would include a justification* $J_2 = (\alpha, [])$ *having an empty support list.*

3.2.4 Belief Revision

Contradictory beliefs were accepted in the Jason's belief base and no belief revision was performed; no preference on beliefs neither. The agent could believe in α and its opposite $\sim \alpha$ at the same time. The extended version integrated the notion of belief's certainty into the belief revision decisions and did not allow belief inconsistency. In case of contradiction, the preference is given for the belief with the higher degree: belief with the smaller certainty degree in the inconsistency pair will be contracted/discarded, and the other belief will be added/kept. In the case of equal certainties, the new belief is given the preference.

A contraction algorithm was proposed: A belief α is not contracted unless a more preferred belief $\sim \alpha$ was added. When contracting a belief α, there is no need to contract beliefs that derived it: when the rule deriving α will attempt to add it again, the addition will be discarded because it will be faced by $\sim \alpha$ that is more preferred. In other terms, the belief in question is contracted with its related justifications without contracting neither the rule's premises nor the rule itself. Beliefs with no justifications will also be contracted.

4 THE KNOWLEDGE-AWARE IR FRAMEWORK

The IR framework proposed in (El Zein and da Costa Pereira, 2020a) consists of a search agent that personalises results to the user's knowledge. The framework considers a client-side agent that uses the content of the documents read by the user to understand his/her knowledge. For every submitted query, the flow is as follows: (*i*) the agent sends the query to the search system and receives a list of candidate documents (ii) the agent examines the content of the documents in the list and measures the similarity of each document with the set of beliefs (iii) the agent returns a filtered list of documents according to the similarity results (iv) the user reads a document (v) the agent considers the content of the document is an acquired knowledge by the user. The keywords representing the examined documents are added as agent beliefs (vi) a reasoning cycle is performed to run the applicable rules and revise the beliefs whenever needed.

The IR agent is modeled as a *Rule-based* entity. When the IR agent has α in its belief base, it believes that the user knows that α is true. If the belief base contains $\neg\alpha$, then the agent believes the user knows that α is not true. When neither α nor $\neg\alpha$ is in the belief base, the agent believes neither the user knows α is true nor the user knows that α is false. The agent also has some knowledge rules that will help it reason and derive news beliefs. During an agent's reasoning cycle, the validity of the rules is checked. A rule is considered valid if all the conditions in the premises are satisfied (the premise exists in the belief base), the rule is fired and the belief in the body is added as a belief. The rules were considered static, and their extraction/origin was not discussed.

The agent acquires its beliefs about the user's knowledge from the documents the user has read. When the user reads a document d, the agent extracts the document's content and considers it an acquired knowledge by the user. The authors in (El Zein and da Costa Pereira, 2020a) applied the *Rapid Automatic Keyword Extraction RAKE* (Rose et al., 2010) as an easy and understandable method, to extract the set of keywords representing the document and calculate their related score. The agent's beliefs will be then represented by the set of keywords extracted with *RAKE*, called *extracted beliefs*.

Knowledge is gradual: an agent might have beliefs more entrenched than others. The "degree" measuring this entrenchment is defined as below:

Definition 3. *The degree of a belief* α *is the degree to which the agent believes the user is knowledgeable about* α*. It is represented by a decimal* $[0;1]$*, where* 0 *means the lowest degree –the agent believes the user has absolutely no knowledge about* α*, and* 1 *means the highest degree –the agent believes the user has the maximum knowledge about* α*.*

A document $d = \{(k_1;s_1),\ldots,(k_n;s_n)\}$ *is a set of tuples where* k_i *is the keyword extracted by RAKE and* s_i *is its related score.* k_i *is associated with an* ***extracted belief*** b_v *whose degree is calculated as follows:*

$$degree(b_v) = \lambda \cdot \frac{s_i}{\max(s_j)_{k_j \in d}} \tag{5}$$

In Equation 5 the RAKE score of an extracted keyword is normalized then multiplied by an adjustment factor $\lambda \in [0,1]$ that weakens the scores' magnitude. The factor's value may vary based on different factors like the source of the document, for example.

This equation allows the calculation of the degree for *extracted beliefs* only. As for *derived beliefs*, their degrees will depend on the degree of premises that derived them. For that reason, the beliefs' dependency is tracked using Justifications and nodes as discussed in Section 2.3.

The filtering process is based on the similarity $Sim(B,d)$ between the agent's set of beliefs $B = \{(b_1;degree(b_1)),\ldots,(b_m;degree(b_m))\}$ and the content of a document $d = \{(k_1;s_1),\ldots,(k_n;s_n)\}$ to be proposed to the user. A similarity measure was proposed considering the degrees of the intersected beliefs and the document knowledge.

$$Sim(B,d) = \begin{cases} \frac{\max\{\sum_{k_i \in d}^{n}[e(B,k_i)) - e(B,\neg k_i)],0\}}{|S|} & \text{if } |S| \neq 0 \\ 0 \text{ otherwise.} & \end{cases} \tag{6}$$

The formula is inspired by the similarity function proposed by Lau *et al.* in (Lau et al., 2004). S is the set of keywords appearing both in d and in B, defined by $S = \{k_i \in d : e(B,k_i) > 0 \vee e(B,\neg k_i) > 0\}$.

The extent $e(B,k_i) = degree(k_i)$ if $k_i \in B$; and 0 otherwise. In simple terms, the similarity formula calculates the average belief degree of the intersected keywords between the document and the beliefs. The result is a value between 0 and 1; where 1 means that all the intersected keywords are strong entrenched beliefs. A similarity 0 means either that the document has no keywords in common with the beliefs or that it contains more contradictory information than similar ones when compared to the beliefs.

The similarity formula "rewards" the documents containing common keywords with the set B and penalizes those containing keywords whose corresponding negated beliefs are in B.

Finally, a cutoff value γ is set for $Sim(B,d)$ that allows deciding whether the knowledge inside a document is similar to a set of beliefs or not. We can cite at least two main applications for this framework: (1) *Reinforcing* the user's knowledge: returning the documents that are "close" to the agent's beliefs (those having a similarity score higher than the cutoff) will be returned. (2) *Novelty*: returning documents with new content with respect to what the user already knows, the documents having similarity below the cutoff will be returned.

5 PROPOSED INFORMATION FILTER AGENT

5.1 System Design

The primary concern of this paper is the development of a proof of concept of an agent-based system in the cognitive IR domain. We develop the IR filter discussed in Section 4 using the extended version of Jason discussed in Section 3.2. The work presented here is, to our knowledge, the first implemented work of information filter agents that considers the user's knowledge. We justify the choice of the extended Jason language to model agents by its ability to model rule-based agents.

Jason's extension considers beliefs as facts, assigns entrenchment degree to them represented by *degOfCert* and deals with belief inconsistency. It also allows the representation of knowledge-rules that will derive new beliefs thanks to the *trigger-independent plans*. In addition, Jason implements the dependency approach proposed by Alechina *et al.* in (Alechina et al., 2005), and used in (Alechina et al., 2006), to track the dependencies between beliefs by associating dependency and justifications lists for each belief. Another particular advantage of Jason is that it is an open-source interpreter written in Java, which makes the development easy and customisable. A further advantage of choosing the Jason extension proposed in (El Zein and da Costa Pereira, 2020b) to implement the IR framework proposed in (El Zein and da Costa Pereira, 2020a) is that both, the framework and the extension, track the beliefs by following the method discussed in Section 2.3.

To perform the proposed integration, we associate the notions of the IR framework to the Jason's language as follows:

- Jason's beliefs are the agent's beliefs about the user's knowledge.
- The entrenchment degree of a belief will be represented by the *degOfCert* in Jason's belief annotation. It will represent the agent's estimation of the user's knowledge regarding a concept.
- *Initial beliefs*: are beliefs that represent initial information about the user. Their degrees are explicitly expressed in the beliefs' annotation. During the first reasoning cycle, for every initial belief, one justification with an empty support list is created.
- *Extracted beliefs*: are keywords extracted from the content of the documents. They will be considered as perceived beliefs and their belief degree will be calculated as per Equation 5.
- *Derived beliefs*: are beliefs that result from firing the applicable rules and from the reasoning process. Their degree is calculated as per Equation 1.
- *Contextual rules*: are considered static and cannot be revised.

The facts represent information that the agent has currently obtained about its user's knowledge. The user's knowledge is represented as Jason beliefs, their related degree is expressed as *degOfCert* tracked in the belief annotation. Those beliefs might change over time as a result of the addition/deletion of other beliefs due to: (i) reading new documents that might contain new information, contradictory information, or redundant information with different degrees, and to (ii) the rule's reasoning process itself, that will derive new beliefs from the agent existing beliefs or delete beliefs in case of inconsistency. To represent a rule $\alpha_1 \& \alpha_2 \rightarrow \beta$ in Jason, the syntax is $+tei$: α_1 & $\alpha_2 \leftarrow \beta$.

To maintain the belief base consistency, the entrenchment degree of beliefs must be raised or lowered via a belief revision operation $K * (\alpha, i)$ where α is a new belief and i is its new entrenchment degree.

We propose the following to revise belief:

$$K * (\alpha, i) = \begin{cases} \text{If } \alpha \notin K: \ K + (\alpha, i) & \\ \text{If } \alpha \in K: & \\ \quad K + (\alpha, i), & \text{if } i > j \\ \quad Nothing, & \text{if } i < j \\ \text{If } \neg\alpha \in K: & \\ \quad K \div (\neg\alpha, j) \ then \ K + (\alpha, i), & \text{if } i > j \\ \quad Nothing, & \text{if } i < j \end{cases} \tag{7}$$

The revision operator checks first if α already exists in the belief base. If it is not in the belief base, it is added with the degree i. If α already exists, the two degrees i and j are compared. When the new degree i is smaller than the existing degree j, the degree of α in the belief base is not changed. When i is higher than the existing degree j, an expansion operation $K + (\alpha, i)$ will be initiated and will increase the degree of α from j to i. The revision operator finally checks if $\neg\alpha$ is already in the belief base. If it already exists with a degree j, the preference will be given to the belief with the higher degree. When i is higher than j, α will have the preference to stay, $\neg\alpha$ must be first contracted (or assigned the lowest entrenchment degree equal to zero for example). Then, α is added with degree i. Finally, when i is smaller than j, the addition of α is discarded.

5.2 Use Case: Application for Novelty

We built the information system framework discussed in Section 4 in Java and modeled the filter agent using the extended version of Jason discussed in Section 3.2. We present in this section a use case of an interaction between a user and the proposed system. The system will be employed to return novel documents with respect to the user's knowledge. We examine the returned results in response to the submitted queries, investigate the knowledge extraction process and discuss the filtering decision.

The IR part is built on top of the open-source library Apache Lucene (luc, 2020) configured with the standard analyser for indexing and for searching. It is set to return 10 documents ranked by relevance to the query. We use a public dataset of short BBC articles (Greene and Cunningham, 2006), select the 400 documents of the technology topic and remove duplicate documents. We rename the text files to include the article's title and finally build the search index.

We choose to set the adjustment factor λ of Equation 5 to be equal to 0.9. The advantage of this factor is that it prevents having the entrenchment degree of the most representative keyword of a document d to be equal to 1. The mentioned keyword is the one having the maximum RAKE score: k_i *where* $k_i \in$

```
// Beliefs

//Plans
+tei : the_psp & gizmondo_store <- +sony.
```

Figure 2: Initial state of Jason filtering agent.

```
1- d363 - Sony PSP handheld console hits US.txt
2- d025 - Sony PSP console hits US in March.txt
3- d359 - Gizmondo gadget hits the shelves.txt
4- d153 - DS aims to touch gamers.txt
5- d185 - Nintendo DS aims to touch gamers.txt
6- d348 - Game makers get Xbox 2 sneak peek.txt
7- d247 - Nintendo adds media playing to DS.txt
8- d134 - Gates opens biggest gadget fair.txt
9- d351 - Nintendo DS makes its Euro debut.txt
10- d174 - Gadgets galore on show at fair.txt
```

Figure 3: The list of documents returned by Lucene for q_1.

d and $s_i = max(s_j)_{k_j \in d}$; when its related entrenchment degree is calculated it is "normalised" by dividing it by itself which results to 1. That would mean if the user has read a document having "Vaccine efficiency" as the most representative keyword, the agent would consider the user already reached the maximum knowledge about vaccine's efficiency, which is not realistic. We set the similarity's cutoff value, introduced in Section 4 to $\gamma = 0.25$. This value is the threshold to the similarity $Sim(B,d)$ between the belief set and the candidate document d. In a novelty context, only documents having a similarity smaller than γ are returned. The closer the cutoff value is to 0 the more conservative the approach and the more novel documents are returned.

We assume at time zero, i.e. before the user-system interaction starts, the agent has no information about the user: the belief set B is empty. The agent has a plan representing a contextual rule that derives the user knows about the *Sony* corporation if he/she knows about both the Gizmondo store and the PlayStation portable *psp*. This plan could originate for example from mining contextual knowledge from textual corpus, like the information flow discussed (Lau et al., 2004). The initial state of the agent is expressed in Figure 2.

The interaction starts by submitting a query q_1"Gaming device". The agent relays the query to Lucene and receives a ranked list of 10 documents responding to the query -displayed in Figure 3. Since the agent has no beliefs about the user's knowledge yet, there is nothing to compare with the content of the documents. In this case, any document's content will be considered novel to the user's knowledge. In consequence, the agent proposes to the user the list of 10 documents without any filter applied.

Table 2: Sample of the keywords extracted from d_{363}.

Belief	RAKE Score	Belief Degree
the_gizmondo_combined_media_player	28.1	0.65
multi_player_gaming	9.3	0.21
gaming_gadget	6.3	0.14
gizmondo_store	5.7	0.13
the_psp	5.3	0.12
ds_handheld	4.7	0.1

Knowledge Extraction and Belief Representation. The user selects for reading the document d_{359} entitled "Gizmondo gadget hits the shelves". The information inside it will be acquired by the user as new knowledge. To represent this knowledge, the agent uses RAKE to extract the keywords representing the document and associates to each of them an entrenched degree. 39 keywords are extracted from the document, some of which are illustrated in Table 2. We have replaced the spaces between words with underscores in order to respect the syntax of Jason's belief. The table shows the RAKE score of some keywords as well as their associated entrenchment degree calculated using Equation 5.

In d_{359}, the keyword "the british-backed gadget faces stiff competition" has the highest RAKE score $\max_{s_j \in d}(s_j) = 38.6$. To normalize the scores of other keywords, their RAKE score will be divided by 38.6 then multiplied by the adjustment factor λ. If we consider for example the belief "gaming_gadget" having a rake score of 6.3, its entrenchment degree is calculated as follows $deg(gaming\_gadget) = 0.9 \cdot (6.3 \div 38.6) = 0.14$. This means that the agent believes the user has some knowledge about *gaming_gadget* with the degree 0.14 on a scale of 0 to 1. In total, 39 expansion operations $B + (\alpha, BeliefDegree)$ are performed to add the new beliefs.

Reasoning and Deriving Beliefs. Once the knowledge is extracted and the beliefs were added, a reasoning cycle is run to fire applicable rules and derive new beliefs if needed. After reading the document d_{359}, the agent plan becomes applicable since its premises conditions are satisfied: the *gizmondo_store* and *the_psp* beliefs are in the belief base. Therefore, the plan is fired and its body gets executed: the belief *sony* is added. To calculate the entrenchment degree of the derived belief, a Justification J_1 gets created with a quality degree equal to $qual(J_1) = \min\{deg(gizmondo\_store), deg(the\_psp)\} = \min\{0.13; 0.12\} = 0.12$. The belief *sony* is then added with $degree(sony) = max\{qual(J_1)\} = 0.12$. $B + (sony, 0.12)$ is performed.

Filtering the Results. The user submits another query q_2 "PSP", Lucene returns to the agent 10 candidate documents displayed in Figure 4. Now that the agent is aware about the user's knowledge, it is capable of filtering the documents according to what the user knows. The similarity $Sim(B,d)$ between the set of beliefs and the content of each of the 10 documents will be measured.

We take for example d_{025} and d_{363}, calculate their similarity with the belief set and examine the related filtering decision. The document d_{025} entitled "Sony PSP console hits US in March" is represented by 19 keywords, out of which 1 is common with the belief set. The set of keywords appearing both in d and in B is $S = \{ds\_handheld\}$. $Sim(B,d_{025}) = \max\{degree(ds\_handheld,0\} \div 1 = max\{0.1,0\} \div 1 = 0.1$. On the other side, d_{363} entitled "sony psp handheld console hits us" has 4 out of 41 common keywords with the belief set. S = {*the_gizmondo_combined_media_* player, gaming_gadget, the_psp, multi_player_gaming} , the $Sim(B,d_{363}) = \max\{0.65+0.21+0.14+0.12,0\} \div 4 = 0.28$. By interpreting these similarities, we conclude that d_{363} is more similar to the belief set (representing the user's knowledge) than d_{025}. In other terms, if the user reads d_{025}, he/she will acquire less novel information compared to d_{363}.

Knowing that the similarity cutoff value is 0.25, 8 documents are returned to the user including d_{025} and excluding d_{363}. The list of returned documents is displayed in Figure 4. Recalling that the aim of this example is to return the documents having novel content: documents having a similar content with the agent's beliefs should be excluded. Notice that d_{359} was returned by Lucene in response to query q_2 but filtered out by the agent because it was already read by the user. The similarity with the belief set is 1.

Belief Expansion. In response to q_2 the user selects to read d_{025}. The 19 keywords representing the document are then added as the user's knowledge: the belief base is revised by 19 keywords $B*(\alpha, BeliefDegree)$ as discussed in Section 2.2. The associated entrenchment degree for *ds_handheld* (the only common keyword with the belief base) is 0.58. When the agent is adding this belief, it notices that the user already has some knowledge about it of 0.1. The agent then increases the related belief degree to 0.58.

A reasoning cycle runs, they are no contradictions to resolve and no plans to fire.

6 CONCLUSION AND FUTURE WORK

This work presented a new use of BDI agents in IR. In the proposed framework, the beliefs of a BDI (Jason) agent have been used to represent the user's knowledge. Besides, the beliefs can be gradual; their related degrees reflect how entrenched is an agent's belief about the user's knowledge regarding a specific topic. The agent also can reason about the user's knowledge, derive new facts, and decide which belief(s) to hold in case of inconsistency. The proposed gives the opportunity of usage in applications requiring personalization and understanding of certain cognitive aspects of the user.

Two of the possible applications of the framework are (1) *novelty*, where the returned documents must contain new information with respect to the user's knowledge and, (2) *knowledge reinforcement*, where documents must contain information that is similar to what the user already knows.

We have also presented an example as a proof of concept of our proposal. We developed the cognitive framework in Java, built the search index with the Lucene library (luc, 2020), integrated it with Jason's extension (El Zein and da Costa Pereira, 2020b) and finally tested it on a dataset of BBC short news (Greene and Cunningham, 2006). The example described a series of real interactions between the user and the proposed framework: the user submits queries to the system, the system responds with personalised documents, the user selects one document to read and possibly would issue another query, and so on. We have also explained how the agent extracted the user's knowledge, represented it as beliefs, and finally used it to "filter" the returned documents. Besides, by interpreting the interaction's result, we have shown how the IR agent can acquire numerous beliefs about concepts that the user is aware of, i.e., for a single document of 500 words, 70 beliefs could be extracted. Notice that those concepts are not equally known by the user; this justifies the allocation of entrenchment degree to each belief. The use case showed how the agent's belief set can "expand" when the user acquires more knowledge (by reading more documents). Besides, it showed how the degree of knowledge about a concept/keyword can increase when the user reads some information that he/she already knows.

Although the developed framework considered the possibility of representing negated beliefs and revising the belief set accordingly, the actual extraction of negated knowledge is a challenging task. The practical research advancement to extract such information from non-structured text remains an unresolved issue in the literature (Blanco and Moldovan, 2011). Therefore, the notion of belief revision could not be tested in a real case scenario.

Another challenge was the calculation of the similarity between the documents' keywords and the set of beliefs. The proposed framework assumes that a

Returned by Lucene	Returned by the agent
1- d025 - Sony PSP console hits US in March.txt	1- d025 - Sony PSP console hits US in March.txt
2- d363 - Sony PSP handheld console hits US.txt	2- d045 - Sony PSP tipped as a 'must-have'.txt
3- d045 - Sony PSP tipped as a 'must-have'.txt	3- d345 - More movies head to Sony's PSP.txt
4- d345 - More movies head to Sony's PSP.txt	4- d238 - Gamers snap up new Sony PSP.txt
5- d238 - Gamers snap up new Sony PSP.txt	5- d351 - Nintendo DS makes its Euro debut.txt
6- d351 - Nintendo DS makes its Euro debut.txt	6- d112 - Portable PlayStation ready to go.txt
7- d112 - Portable PlayStation ready to go.txt	7- d247 - Nintendo adds media playing to DS.txt
8- d247 - Nintendo adds media playing to DS.txt	8- d084 - Nintendo handheld given Euro date.txt
9- d084 - Nintendo handheld given Euro date.txt	
10- d359 - Gizmondo gadget hits the shelves.txt	

Figure 4: The list of documents returned for q_2 before and after filtering.

keyword and a belief are "similar" only if they are literally the same. For example, "multiplayer_gaming" and "multi_player_gaming" are considered two different keywords. We believe that this could be overcome by applying some normalisation or standardisation techniques that are publicly available like NLTK (Bird et al., 2009) and Stanford Core NLP (Manning et al., 2014). Furthermore, the method used to extract the keywords does not take into account their semantics, nor does the similarity formula which compares all the keywords. This task might be more challenging as it requires the integration of some Natural Language Processing techniques. For future work, we plan to consider enhancing the developed framework by extracting normalised keywords with the possibility of semantically comparing them.

REFERENCES

(2020). Apache lucene. https://lucene.apache.org/.

(2021). Jason agent programming. http://jason.sourceforge.net/wp/.

Alchourrón, C. E., Gärdenfors, P., and Makinson, D. (1985). On the logic of theory change: Partial meet contraction and revision functions. *The journal of symbolic logic*, 50(2):510–530.

Alechina, N., Bordini, R. H., Hübner, J. F., Jago, M., and Logan, B. (2006). Belief revision for agentspeak agents. In *AAMAS*, pages 1288–1290. ACM.

Alechina, N., Jago, M., and Logan, B. (2005). Resource-bounded belief revision and contraction. In *International Workshop on Declarative Agent Languages and Technologies*, pages 141–154. Springer.

Bakos, J. Y. (1997). Reducing buyer search costs: Implications for electronic marketplaces. *Management science*, 43(12):1676–1692.

Bird, S., Klein, E., and Loper, E. (2009). *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.".

Blanco, E. and Moldovan, D. (2011). Some issues on detecting negation from text. In *Twenty-Fourth International FLAIRS Conference*.

Bordini, R. H., Hübner, J. F., and Wooldridge, M. (2007). *Programming Multi-Agent Systems in AgentSpeak Using Jason (Wiley Series in Agent Technology)*. John Wiley & Sons, Inc., Hoboken, NJ, USA.

Carrillo-Ramos, A., Gensel, J., Villanova-Oliver, M., and Martin, H. (2005). Pumas: a framework based on ubiquitous agents for accessing web information systems through mobile devices. In *Proceedings of the 2005 ACM symposium on Applied computing*, pages 1003–1008.

Culpepper, J. S., Diaz, F., and Smucker, M. D. (2018). Research frontiers in information retrieval: Report from the third strategic workshop on information retrieval in lorne (swirl 2018). In *ACM SIGIR Forum*, volume 52, pages 34–90. ACM New York, NY, USA.

da Costa Móra, M., Lopes, J. G. P., Vicari, R. M., and Coelho, H. (1998). Bdi models and systems: Bridging the gap. In *ATAL*, pages 11–27.

El Zein, D. and da Costa Pereira, C. (2020a). A cognitive agent framework in information retrieval: Using user beliefs to customize results. In *The 23rd International Conference on Principles and Practice of Multi-Agent Systems*.

El Zein, D. and da Costa Pereira, C. (2020b). Graded belief revision for jason: A rule-based approach. In *International Joint Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT'20)*.

Garcin, F., Zhou, K., Faltings, B., and Schickel, V. (2012). Personalized news recommendation based on collaborative filtering. In *2012 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology*, volume 1, pages 437–441. IEEE.

Gärdenfors, P. and Makinson, D. (1988). Revisions of knowledge systems using epistemic entrenchment. In *Proceedings of the 2nd conference on Theoretical aspects of reasoning about knowledge*, pages 83–95.

Greene, D. and Cunningham, P. (2006). Practical solutions to the problem of diagonal dominance in kernel document clustering. volume 148, pages 377–384.

Guttman, R. H. and Maes, P. (1998). Agent-mediated integrative negotiation for retail electronic commerce. In *International Workshop on Agent-Mediated Electronic Trading*, pages 70–90. Springer.

Jensen, A. S. and Villadsen, J. (2015). Plan-belief revision in jason. In *ICAART (1)*, pages 182–189. SciTePress.

Kurumatani, K. (2004). Multi-agent for mass user support. *Lecture Notes in Artificial Intelligence (LNAI)*, 3012.

Lau, R. Y., Bruza, P. D., and Song, D. (2004). Belief revision for adaptive information retrieval. In *Proceedings* *of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 130–137.

Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J. R., Bethard, S., and McClosky, D. (2014). The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*, pages 55–60.

Rao, A. and Georgeff, M. (2001). Modeling rational agents within a bdi-architecture.

Ricci, F., Rokach, L., and Shapira, B. (2011). Introduction to recommender systems handbook. In *Recommender systems handbook*, pages 1–35. Springer.

Rose, S., Engel, D., Cramer, N., and Cowley, W. (2010). Automatic keyword extraction from individual documents. *Text mining: applications and theory*, 1:1–20.

Williams, M.-A. (1995). Iterated theory base change: A computational model. pages 1541–1549.

Yu, R., Tang, R., Rokicki, M., Gadiraju, U., and Dietze, S. (2021). Topic-independent modeling of user knowledge in informational search sessions. *Information Retrieval Journal*, 24(3):240–268.