IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received November 11, 2016; accepted November 21, 2016, date of publication December 20, 2016, date of current version February 25, 2017.

Digital Object Identifier 10.1109/ACCESS.2016.2642402

Machine Learning and Conceptual Reasoning for Inconsistency Detection

JAMEELA AL OTAIBI, ZEINEB SAFI, ABDELÂALI HASSAÏNE, FAHAD ISLAM, AND ALI JAOUA
Computer Science and Engineering Department, College of Engineering, Qatar University, Doha 2713, Qatar

Corresponding author: A. Jaoua (jaoua@qu.edu.qa)

This work was supported by the NPRP Grant from the Qatar National Research Fund (a member of Qatar Foundation) under Grant 06-1220-1-233. The statements made herein are solely the responsibility of the authors.

ABSTRACT This paper focuses on detecting inconsistencies within text corpora. It is a very interesting area with many applications. Most existing methods deal with this problem using complicated textual analysis, which is known for not being accurate enough. We propose a new methodology that consists of two steps, the first one being a machine learning step that performs multilevel text categorization. The second one applies conceptual reasoning on the predicted categories in order to detect inconsistencies. This paper has been validated on a set of Islamic advisory opinions (also known as fatwas). This domain is gaining a large interest with users continuously checking the authenticity and relevance of such content. The results show that our method is very accurate and can complement existing methods using the linguistic analysis.

INDEX TERMS Information extraction, conceptual reasoning, text categorization, hyper rectangular decomposition, inconsistency detection.

I. INTRODUCTION

Detecting inconsistencies is widely used to ensure that data are accurate and appropriately integrated within large textual corpora. It is indeed important nowadays to have reliable methods in order to check the integrity of continuously increasing text-based content. In order to be accurate, those methods need to be fine-tuned for each domain, as a more general method is likely to be less efficient. Unfortunately, existing methods for inconsistency detection use generally complicated linguistic analysis of text that does not yield efficient results. The main reason is that linguistic analysis needs more research investigation, but also because existing methods tend to work on a large domain and are therefore less accurate. On the other side, trying to detect inconsistency in a text without defined targeted potential inconsistencies should not give accurate results. In order to tackle this problem, we propose a new method based on multilevel classification and conceptual reasoning with expected targets. The proposed approach detects the domain and all sub-domains of any content before performing the inconsistency detection using conceptual reasoning. We validated the method on the trending topic of Islamic advisory opinions called Fatwas. Detecting inconsistencies in this domain is important, as it will allow users to detect unreliable content. Entities using such kind of opinion can use it to check textual data integrity and avoid some omissions or ambiguities. If two seemingly identical user inquiries are given conflicting judgments by scholars, this should be detected as inconsistency. The remainder of this paper is structured as follows: Section II illustrates the work found in the literature related to inconsistencies detection. Section III describes the basic definitions for multilevel classification and ConProve tool as defined in [1]. Section IV describes the database used in this study. Section V presents a description of our approach, Section VI demonstrates and discusses the results and Section VII concludes this work and gives insights for future research.

II. RELATED WORK

Inconsistency detection is an active research field with many applications. In this section, we review some of the recent published methods in this field. Tpper *et al.* [2] proposed a new method for DBpedia ontology enrichment for inconsistency detection based on the enrichment of the ontology using statistical methods. Hollenstein proposed a method for inconsistency detection within completed annotations. Hollenstein *et al.* [3] proposed a discrepancy ranking and an entropy ranking and show that the method can be applied efficiently for inconsistency detection. Gutierrez proposed an Ontology-based Error Detection in Text. Gutierrez [4] argues that such a method can provide useful insight that aims at determining incorrect text and specific axioms that cause the contradiction. Bannay *et al.* proposed an innovative method based on both natural language processing and logical satisfiability checking in order to detect inconsistencies in text. The authors claim that the method can also check for redundancy and incompleteness in procedural texts. We can also use it for logically translating technical texts [5]. Tan *et al.* proposed a method for detecting inconsistencies within comments associated to Javadoc. The authors infer a set of likely properties by analyzing the comments associated to Java methods. In a second step, they generate random tests for these methods in order to compare them against the inferred properties and thus checking inconsistencies [6]. Motte *et al.* [7] proposed an invention that uses named entity extraction and comparison in order to detect inconsistency between News documents. Van der Aa *et al.* proposed a method for inconsistency detection between textual and model-based process descriptions. The approach is based on natural language processing followed by similarity computation and alignment steps [8]. Carmeli *et al.* proposed a method for detecting inconsistencies between structured and unstructured contents. The method has mainly been developed for electronic health records and applies both natural language processing and machine learning to detect inconsistencies [9]. Pariyar *et al.* proposed a rule based method for detecting inconsistencies in Multilanguage knowledge sharing platforms. The method has been validated on data from Wikipedia and the author claims it can help natural language processing based systems in detecting inconsistencies at early stages [10]. Xiao *et al.* proposed a method called text2policy for extracting access control policies (ACP) from requirement documents as well as action steps from use cases. Once extracted, those ACPs are easily checkable for inconsistencies [11]. We presented a method based on named entity extraction and conceptual reasoning for detecting inconsistencies between fatwas. The method was mainly based on natural language processing techniques and did not achieve accurate results [12]. By an analysis of the state of art for inconsistency detection, we can see that most existing methods deal with the problem of inconsistency detection by using deep linguistic text analysis, which happens to not be very accurate. Ontology based inconsistency detection has the advantage to prepare for consecutive text classification through the most relevant path in the tree. Added to this, by splitting the set of labels to two subsets: premises and consequences, we have been able to simplify the problem and obtain general good results in the domain of Fatwas. The method is applicable for any organized domain of texts where we may classify labels (i.e. category of the label of the method) to be either premises or consequences. In our case, all labels of the domain structure represent potential premises of some rule, except labels associated to the leaves of the tree which correspond to judgments representing potential consequences. By using a machine learning approach, we discover a path of labels associated to any text. By this way, we derive a logical rule for each text induced by the associate path. By comparing different generated rules associated to different texts, a logical prover may discover potential inconsistencies. In this study, we propose a new approach that tackles this problem by introducing a multilevel classification tree approach for rules generation (i.e. path of labels associated to a text) followed by a conceptual reasoning approach to discover inconsistencies.

III. THEORETICAL FOUNDATIONS

In section III-A, we present a set of basic definitions related to the field of Formal Concept Analysis [13]. These definitions constitute the foundation for our hyper conceptual keyword extraction method presented in section III-B and the conceptual reasoning tool called ConProve presented in section III-C.

A. BASIC DEFINITIONS

Definition 1: *A relation from a set O to a set P is a subset of the Cartesian product $O \times P$.*

Definition 2: *Given O as our set of objects and P as our set of attributes, $\mathcal{R}$ is a relation on $O \times P$. Relation $\mathcal{R}(o, p)$ holds if an object $o \in O$ has attribute $p \in P$. This triplet $K = (O; P; \mathcal{R})$ is called a formal context.*

In our context, the set of objects O consists of documents, the set of attributes P consists of a set of keywords and the relation $\mathcal{R}$ represents whether a given document o contains a certain keyword p or a representative stem of p.

Definition 3: *The image of an object o given by the relation $\mathcal{R}$ is defined as $o.\mathcal{R} = \{p \in P|(o, p) \in \mathcal{R}\}$.*
The image of a set O given by the relation $\mathcal{R}$ is defined as $O.\mathcal{R} = U_{o\in O}\{p \in P|(o, p) \in \mathcal{R}\}$.

Definition 4: *The composition (relative product) of two relations $\mathcal{R}$ and $\mathcal{R}'$ is given by: $\mathcal{R} \circ \mathcal{R}' = \{(u, v)|\exists w|((u, w) \in \mathcal{R})$ and $((w, v) \in \mathcal{R}')\}$.*

Definition 5: *The converse of the relation $\mathcal{R}$ is $\mathcal{R}^{-1} = \{(p, o)|(o, p) \in \mathcal{R}\}$*

Definition 6: *The binary relation $\mathcal{I}(A)$ on a set A (i.e., a subset of $A \times A$) is called the identity relation. It is defined as: $\forall p \in A, p.\mathcal{I}(A) = \{p\}$.*

Definition 7: *The number of pairs in $\mathcal{R}$ is called cardinality of $\mathcal{R}$: $Card(\mathcal{R}) =$ number of pairs $(o, p) \in R$.*

Definition 8: *"Assume two partially ordered sets, $(A, \leq_A)$ and $(B, \leq_B)$. Let $f : A \to B$ and $g : B \to A$ such that $\forall a \in A, b \in B, f(a) \leq_B b \Leftrightarrow g(b) \leq_A a$. Therefore, (f, g) is called a Galois connection.*
Let there be two arbitrary sets O and P and $\mathcal{R}$ is a Relation on $O \times P$. Let A and B be two arbitrary sets such that $A \subseteq O$ and $B \subseteq P$. (f, g) is a pair of functions where $f : 2^O \to 2^P$ and $g : 2^P \to 2^O$ defined by:

- $f(A) = \{p \in P|\forall o \in A, (o, p) \in R\}$
- $g(B) = \{o \in O|\forall p \in B, (o, p) \in R\}$,

forms a Galois connection. [14]"

Definition 9: *"The pair (f, g) is called an Extended Galois connection.*
Let there be two arbitrary sets O and P and $\mathcal{R}$ is a Relation on $O \times P$. Let A and B be two arbitrary sets such that $A \subseteq O$ and $B \subseteq (P \cup P')$. P' is defined as the set of negation of element P. For example, if $P = \{a, b\}$ then $P\cdot = \{\neg a, \neg b\}$.

Remark: By definition, as in logic $\neg\neg(p) = p$.
The pair of functions (f, g) with $f : 2^O \rightarrow 2^{P \cup P'}$ and $g : 2^{P \cup P'} \rightarrow 2^O$ defined by:

- $f(A) = \{p \in P | \forall o \in A, (o, p) \in \mathcal{R}\} \cup \{\neg p | p \in P | \forall o \in A, (o, p) \notin \mathcal{R}\}$,
- $g(B) = \{o \in O | \forall p \in (B \cap P), (o, p) \in \mathcal{R}$ and $\forall p \in (B \cap P'), (o, \neg p) \notin \mathcal{R}\}$

forms an extended Galois connection. [14]"

Definition 10: " We call $(g \circ f)(A)$ the closure of A, and $(f \circ g)(B)$ the closure of B. The pair (A, B), where $A \subseteq O$, $B \subseteq P$, $f(A) = B$ and $g(B) = A$ is called a formal concept of context K with extent A and intent B. [14]"

Definition 11: "Let $K = (O, P, \mathcal{R})$ be a formal context and $p \in P$ an attribute. We call the hyper rectangle (or hyper concept) associated to attribute p and denoted by $H_p(\mathcal{R})$ the sub relation of $\mathcal{R}$ such that: $H_p(\mathcal{R}) = \mathcal{I}(p.\mathcal{R}^{-1}) \circ R$. [15]"

Remark 1: $H_p(\mathcal{R})$ is the union of all formal concepts containing p.

Definition 12: The hyper rectangle $H_p(\mathcal{R})$ has a weight $W(H_p(\mathcal{R}))$:
$W(H_p(\mathcal{R})) = \frac{r}{d*c} * (r - (d + c))$,
Where r (respectively d and c) is the cardinality of $H_p(\mathcal{R})$ (respectively its domain and its codomain).

Remark 2: The factor $\frac{r}{d*c}$ measures the hyper rectangle density, while $(r - (d + c))$ measures the space economy obtained if the hyper concept is only approximately represented by its domain and co-domain [16].

Definition 13: The hyper rectangle that has the highest weight is called the optimal hyper rectangle $maxH_p(\mathcal{R})$:
$W(H_p(\mathcal{R})) \geq W(H_b(\mathcal{R}))), \forall b \neq p, b \in COD(\mathcal{R})$,
Where $COD(\mathcal{R})$ is the set of images in $\mathcal{R}$.

Definition 14: We obtain the Remaining Binary Relation R_m by subtracting the optimal Hyper Rectangle from the original relation $\mathcal{R}$:
$R_m(\mathcal{R}) = \mathcal{R} - maxH_p(\mathcal{R})$.

B. HYPER RECTANGULAR KEYWORD EXTRACTION

The hyper rectangular keywords extraction method [17], [18] works as follow:

- A formal context is constructed from the document corpus where the documents form the set of objects O, the words contained in all documents form the set of attributes P and the relation $\mathcal{R}$ indicates whether or not a term is contained in a. Figure 1 shows the formal context that can be built from a text corpus.
- After the context is constructed, we proceed to extract the hyper concept that is associated with every attribute (keyword) which is by definition the union of all formal concepts containing the given keyword.
- The weight of each hyper concept is computed as indicated in definition 12. Figure 2 shows hyper concepts extracted from a corpus of four documents and four terms, and the corresponding weight of each hyper concept.

Co-domain

Domain

	$Word_1$	$Word_2$	…	$Word_n$
$Document_1$	1	1	…	0
$Document_2$	0	1	…	1
⋮	⋮	⋮	⋮	⋮
$Document_M$	0	1	…	0

FIGURE 1. Formal context from corpus.

c=card(codomain)=4

d=card(domain)=4

r=card(relation)=11

	$word_1$	$word_2$	$word_3$	$word_4$
Doc_1	1	1	0	0
Doc_2	1	0	1	1
Doc_3	1	1	1	0
Doc_4	0	1	1	1

$W(H(R)) = \frac{r}{d*c} * (r - (d + c))$

	$word_1$	$word_2$	$word_3$	$word_4$
Doc_1	1	1	0	0
Doc_2	1	0	1	1
Doc_3	1	1	1	0

$W(H(R)) = \frac{8}{3*4} * (8 - (3 + 4))$
$= 0.66$

	$word_1$	**$word_2$**	$word_3$	$word_4$
Doc_1	1	1	0	0
Doc_3	1	1	1	0
Doc_4	0	1	1	1

$W(H(R)) = \frac{8}{3*4} * (8 - (3 + 4))$
$= 0.66$

	$word_1$	$word_2$	**$word_3$**	$word_4$
Doc_2	1	0	1	1
Doc_3	1	1	1	0
Doc_4	0	1	1	1

$W(H(R)) = \frac{9}{3*4} * (9 - (3 + 4))$
$= 1.5$

	$word_1$	$word_2$	$word_3$	**$word_4$**
Doc_2	1	0	1	1
Doc_4	0	1	1	1

$W(H(R)) = \frac{6}{2*4} * (6 - (2 + 4))$
$= 0$

FIGURE 2. Hyper concept from corpus.

- The optimal hyper concept (the hyper concept with the highest weight) is extracted and the remaining relation is computed by subtracting the hyper concept from the original relation as shown in Figure 3.
- An optimal new hyper concept and an optimal remaining relation are extracted from the current remaining relation until the full relation is covered. The hyper concepts extracted so far will form the first depth of the tree.
- For each hyper concept in the first depth, the same process is repeated until a hyper concept tree is constructed.

Remark 3: In this paper we deal with two notions that should not be confused: The tree of the hyper concepts and the multilevel classification tree. In order to distinguish between the two, we use throughout this paper the word depth when we talk about the different levels of the hyper concept tree and the word level when we talk about the multilevel trees of categories.

Original relation

	$word_1$	$word_2$	$word_3$	$word_4$
doc_1	1	1	0	0
doc_2	1	0	1	1
doc_3	1	1	1	0
doc_4	0	1	1	1

-

Optimal hyper rectangle

	$word_1$	$word_2$	$word_3$	$word_4$
doc_2	1	0	1	1
doc_3	1	1	1	0
doc_4	0	1	1	1

=

Remaining relation

	$word_1$	$word_2$
doc_1	1	1

FIGURE 3. Remaining relation from optimal hyper concept.

TABLE 1. ConProve symbols.

Component	Symbol	ConProve representation
Term	Letters	Letters
Negation	$\neg$	!
Conjunction	$\wedge$	&&
Disjunction	$\vee$	||

C. INCONSISTENCIES DETECTION USING ConProve

The second important component of our solution is the ConProve tool [1], [14]. Consequently, it is necessary to briefly explain the theoretical background behind it. ConProve is a propositional logic prover. Given a propositional formula, ConProve decides whether or not a goal holds. Table 1 presents some propositional logic symbols and their ConProve equivalent. The implication operation does not have a ConProve equivalent. Any formula of the form $a \rightarrow b$ should be converted to $!a||b$.

TABLE 2. TTBR for $!a||b$.

	a	b
s1	1	1
s2	0	1
s3	0	0

Each term in a propositional formula can have a truth value of either true or false. Based on the values given to its terms, a propositional formula can have a value of either true or false. A truth table binary representation (TTBR) can be created for each propositional formula, which enumerates all possible term assignments that will make the expression true. In ConProve, TTBR is represented as a formal context $K = (S; T; R)$ where the set of objects S represents the solutions, the set of attributes T represents the terms and the relation R represents the truth value given to a term by each possible solution. Table 2 shows the TTBR of the implication rule $a \rightarrow b$ which is represented in ConProve as $!a||b$.

Figure 4 shows a diagram of the operation steps of ConProve.

- First, the negation of the goal formula is generated and added to the set of input formulas.
- The reduced truth table corresponding to each formula is constructed by considering the initial facts and applying the extended Galois connection of the intermediate TTBR. Extended Galois connection involves either positive or negative properties as shown in definition 9.
- ConProve constructs the join TTBR. Two TTBRs T_1 and T_2 are joined by checking the inconsistency between each row of T_1 with each row of T_2. If the two rows are consistent, they are combined and appended into the resulting table, if not they are discarded [1]. This operation is similar to the equi-join operator used in relational database theory [19].
- Once the TTBR constructed, ConProve concludes whether the goal holds or not, and displays the results to the user.

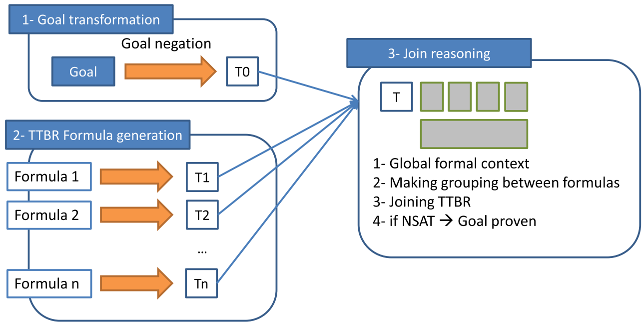

FIGURE 4. ConProve steps.

The Noble Quran	Nobel Hadeeth
Islamic Creed	Religions, Sects and Da'wah (Call to Islam)
Seerah (Biography of the Prophet)	Merits and Virtues
Etiquettes, Morals, Thikr an Du'aa'	Tahaarah (Ritual Purity)
Salah (Prayer)	Funeral: Prayer and Rulings
Zakaah (Obligatory Charity)	Siyaam (Fasting)
Hajj and 'Umrah	Fiqh of Transactions and Inheritance
Women and Family	Foods, Drinks, Clothes and Adornment
Jinaayaat (Criminology) and Islamic Judicial System	Islamic Politics and International Affairs
Medical Issues, Media, Culture and Means of Entertainment	Miscellaneous

FIGURE 5. Categories of Fatwa.

IV. DATABASE DESCRIPTION

This work has been validated on a Fatwa database crawled from Islamweb.net which is an Islamic website that has different attributes that the users can benefit from, such as voice records, prophets sayings, and fatwas, which is our main database in this paper [15]. This database contains about 150,000 '''Islamic advisory opinions''' (aka. fatwas) ranging into more than 20 categories as shown in Figure 5.

Figure 6 shows the hierarchy of categories and subcategories of fatwas, which represent the domains and subdomains for inconsistency detection in our context. A given text is categorized into either a fatwa or not fatwa. The second level of the tree (shown in blue) is the fatwa category. There are more than 20 categories of fatwas but only the most common ones are shown in the tree. Each category of fatwa has some subcategories. The third level shows the three subcategories of Etiquettes related to dealing with others and Fiqh of worship fatwas and the four subcategories of vow fatwas. In this study, we are particularly interested in fatwas related to vow. In our database, this category contains around 1000 Fatwas covering four different subcategories. The last

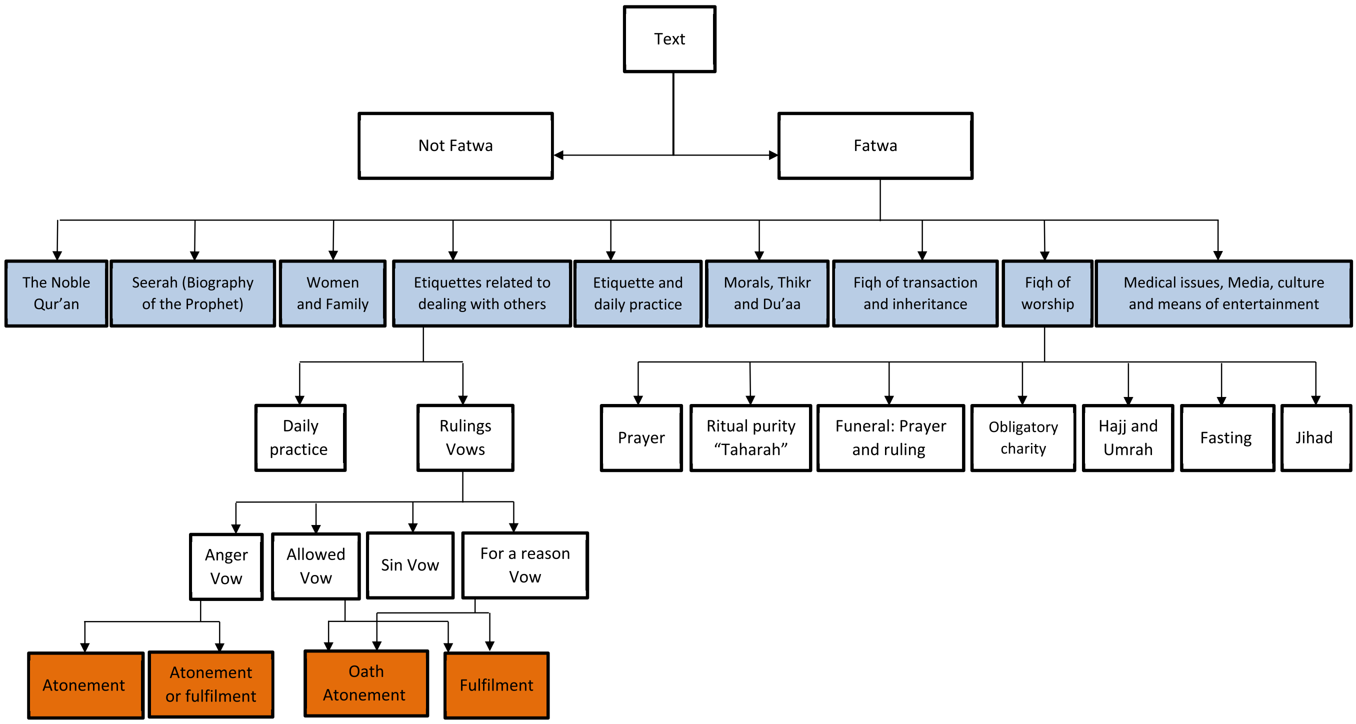

FIGURE 6. Fatwa hierarchy.

level of the tree (colored in orange) shows the Judgments for each subcategory of the vow fatwa which has four values as follows: Atonement, Atonement or Fulfillment, Oath Atonement and Fulfillment.

Since the last level is the one which corresponds to judgment, we can say that the inconsistency occurs between two fatwas when theirs paths of categories and all subcategories are identical but their corresponding judgments are different. This will be our target for inconsistency detection. The following example illustrates the concept.

Assuming we have the following three fatwas:

Fatwa1: "I swore an oath when I was angry not to commit a sin, broke the oath and returned to the sin" and the judgment is "you need to do Atonement"

Fatwa2: "I swore not to talk to my sister while I was angry, but after one day I forgot and talked to her" and the judgment is "you need to do Atonement"

Fatwa3: "I was upset and made an oath not to go to some place, but then I went after one month" and the judgment is "you need to do Atonement or fulfillment"

The fatwas follow the following path in the fatwa hierarchy tree of Figure 6, "Fatwa", "Etiquette related to dealing with others", "Ruling Vows", "Anger Vow" and their corresponding judgments.

Figure 7 shows the path followed by each fatwa starting from "Rulings Vows" labeled A, "Anger Vow" labeled B, and the judgments "atonement or fulfillment" and "atonement" labeled J1 and J2 respectively.

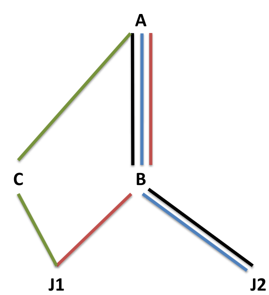

FIGURE 7. Consistent and inconsistent paths.

The path followed by fatwa1, shown in black ($A, B \rightarrow J2$) is consistent with the path followed by fatwa2 shown in blue ($A, B \rightarrow J2$), this is because the same premise (A, B) led to the same conclusion ($J2$). The path followed by fatwa3 in red is inconsistent with both fatwa 1 and 2 ($A, B \rightarrow J1$), this is because the same premise, (A, B) led to a different conclusion ($J1$).

Two fatwas with different premises are consistent even if they share the same conclusion. Assuming we have Fatwa4 shown in green that has a different premise that fatwa3 (A, C) but has the same conclusion ($J1$) fatwa3 and fatwa4 are consistent.

V. METHOD DESCRIPTION

Our proposed method uses a multilevel classifier. It works by first classifying each fatwa to its corresponding category as shown in Figure 8. The figure shows a formal context

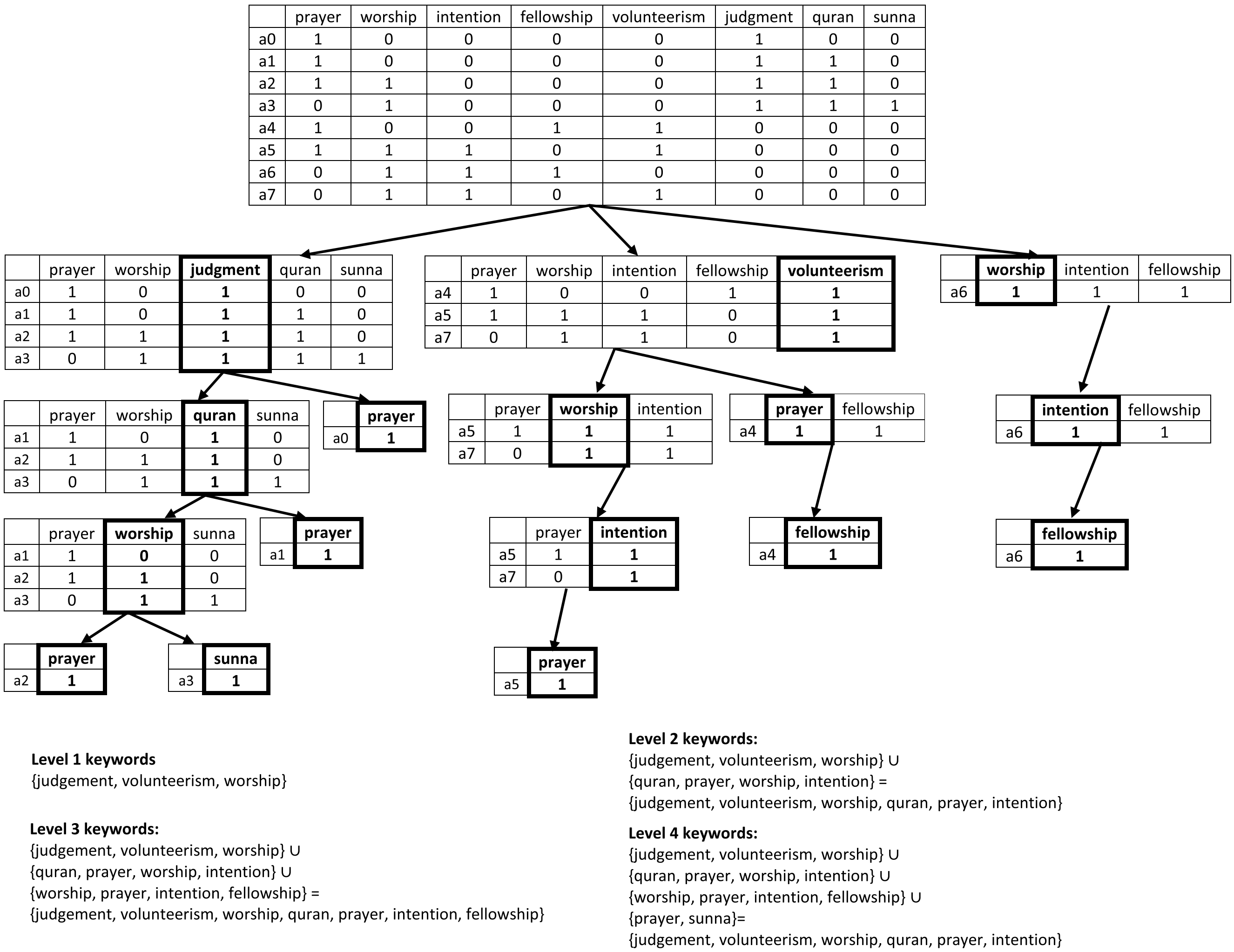

FIGURE 8. **Hyper concept keywords tree.**

that is covered by the corresponding hyper concepts. The keywords at each level are extracted using the hyper concept keyword extraction method introduced in [17] and explained above. These keywords are then fed into a classifier to predict their categories. A propositional logic prover is used to detect inconsistencies. Each fatwa is represented by a propositional logic formula. If the goal is proved, the two compared fatwas are considered consistent, otherwise they are considered inconsistent. The following subsections provide a more detailed explanation of the introduced approach.

A. KEYWORD EXTRACTION AND CLASSIFICATION

Keywords are extracted from the fatwa using the hyper rectangular decomposition algorithm previously defined. The extracted keywords are organized in a tree, and keywords are extracted from each level of the hyper concept tree as shown in Figure 8.

The extracted keywords are passed into a classifier as features to predict the correct category of each fatwa. The classifier used for this experiment is the Random Forest classifier [20], [21]. Our experiment was conducted on the category of Vow related fatwa. The text of the fatwa goes through different classifiers at different levels of the classification tree. The next classifier is decided based on the output label given to the text at the current level. Depending on the classification results, each fatwa is assigned a list of labels. The labels are used in the next step to construct the propositional formula. Classification is performed for the following three levels:

- In the first level, the fatwa is classified into vow-related and non-vow-related fatwas. There are 902 which are vow related and 952 which are not.
- In the second level, the vow fatwa is classified into the four vow categories: Anger related vows (58 documents), vows done for a reason (646 documents), allowed vows (53 documents), invalid vows (140 documents).

- In the third level, the anger related vow fatwas are classified into those that require atonement (9 documents) and those that give the choice between atonement and fulfillment (49 documents).

B. INCONSISTENCY DETECTION USING PROVER ConProve

The labels that are given to the fatwa at each classification level are used to construct the propositional formula. For example, if the fatwa was classified as Vow related → Anger Fatwa → Fulfillment. The corresponding formula will be $V\&\&A\&\&J$ where '"V"' is the symbol for vow related fatwa, '"A"' is the symbol for anger fatwa and '"J"' indicates the judgment of the fatwa which in this case is fulfillment. A second fatwa that is a vow related, anger vow fatwa, but has a different judgment would be represented by the formula $V\&\&A\&\&!J$. By placing the formula corresponding to the first fatwa in the formula section of the tool and the formula corresponding to the second fatwa in the goal section, if both fatwas have the same category and the same judgment, the goal will be proven and the two fatwas are consistent. Otherwise, if both fatwas belong to the same category but have different judgments then the goal will not be proven.

VI. RESULTS

We conducted our experiments on 1854 fatwas out of which 902 are vow related and 952 are not. In the subsequent subsections we present the classification results corresponding to each level of the multilevel classifier and the results of the inconsistency detection using ConProve.

A. CLASSIFICATION RESULTS

In the first level of the classification, we classify each fatwa to either vow or non-vow related. We used 70% of the fatwas for training and the remaining 30% for testing. The keywords obtained from the hyper concept tree at depth 22 were fed to a random forests classifier and the obtained accuracy was 99%. In the second level of the multilevel classification, we classify the vow fatwa into one of the 4 previously mentioned categories. The best classification performance was achieved using keywords of depth 8 of the hyper concepts tree. We took 10 random splits of 70% of the fatwas for training and 30% for testing; the average accuracy obtained was 77.76%. In the third classification level, we are only concerned with anger related fatwas. There are two possible judgments for this type of fatwas. The judgment is either to request atonement or giving the option of atonement or fulfillment. At this level, the class of fatwas that are giving the option of both atonement and fulfillment is over represented. Consequently, we use the F1-measure instead of the accuracy to measure the classifiers performance as it is more suitable for unbalanced datasets. Figure 9, Figure 10, and Figure 11 show the performance of the first, second and third levels of the multilevel classification respectively versus the depth of the hyper concept tree.

The classification results are stored as strings of labels for each fatwa and are used for inconsistency detection, the results of which are reported in the next section.

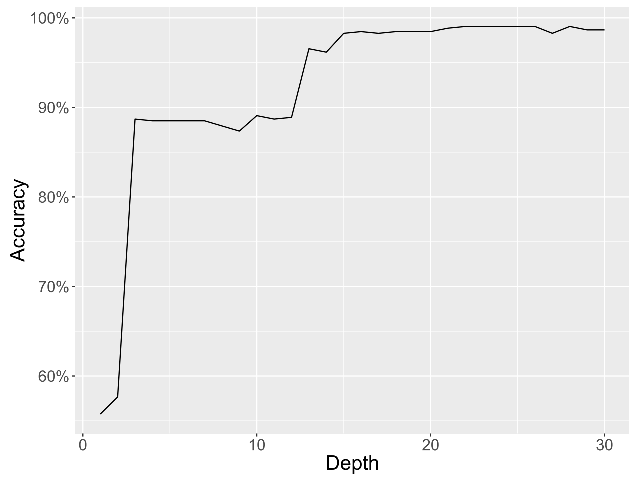

FIGURE 9. First level classification performance.

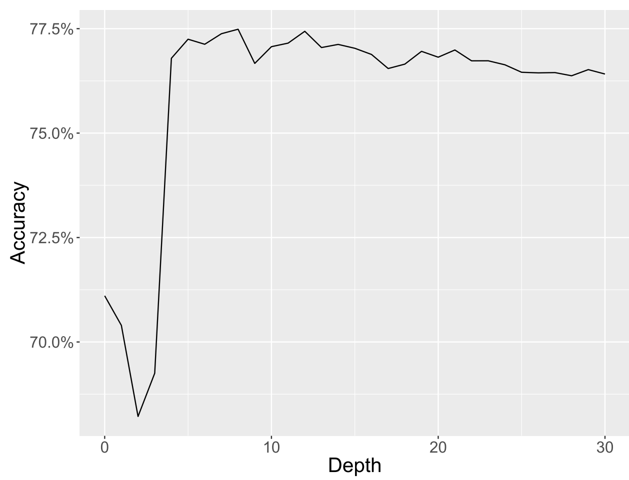

FIGURE 10. Second level classification performance.

B. INCONSISTENCY DETECTION RESULTS

The strings of labels of each fatwa are fed to ConProve to detect inconsistencies between pairs of fatwas. There are four potential cases after following these steps: The two fatwas are consistent and detected as consistent (true positive). The two fatwas are inconsistent and detected as inconsistent (true negative). The two fatwas are consistent but detected as inconsistent (false negative). The two fatwas are inconsistent but detected as consistent (false positive).

By counting each of those instances, the following measures are computed:

- $Precision = \dfrac{\#true\ positives}{\#true\ positives + \#true\ negatives}$
- $Recall = \dfrac{\#true\ positives}{\#true\ positives + \#false\ negatives}$
- $F1\ measure = \dfrac{2 * Precision * Recall}{Precision + Recall}$

Figure 12 illustrates the inconsistency detection performance for increasing depth of the hyper rectangle tree. Results are averaged over 10 random splits of the data into 70% training and 30% testing.

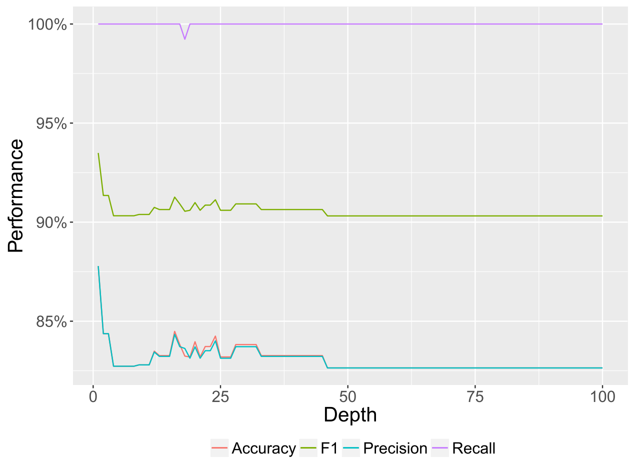

FIGURE 11. **Third level classification performance.**

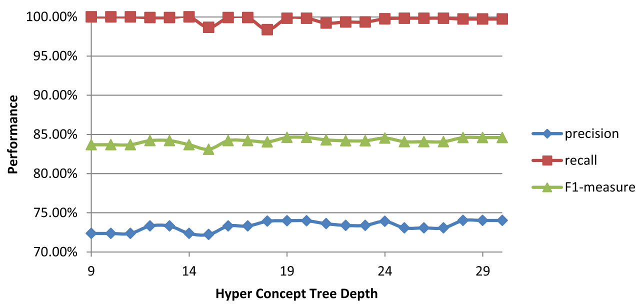

FIGURE 12. **Fatwa Inconsistency detection performance for increasing depth of the hyper rectangle tree.**

C. DISCUSSION AND ANALYSIS

From the obtained results, we can see the following:

- The hyper conceptual feature extraction algorithm successfully extracts keywords that are discriminative as they produce accurate results when fed to the classifier.
- The performance generally increases with the first depths of the hyper concept tree and then either remains constant or slightly decreases which means that our algorithm extracts the most discriminative features in its first depths.
- The optimal inconsistency detection performance is about 85% which confirms that our machine learning method is more accurate than the one based on linguistic analysis [12].

VII. CONCLUSION AND FUTURE WORK

This study suggests that inconsistency detection can achieve high performance when done using multilevel text categorization followed by conceptual reasoning. The keyword extraction method based on hyper rectangular decomposition shows that this method successfully extracts discriminative keywords which are representative of each category or subcategory of documents. The method has been validated on a set of Islamic advisory opinions, but can be generalized to other domains. Future work includes the comparison as well as the combination of our method with other linguistic analysis based methods. Validation on larger datasets is also to be considered. Finally, the use of other classification and clustering methods is to be studied as well.

REFERENCES

[1] S. Elloumi, B. Boulifa, A. Jaoua, M. Saleh, J. A. Otaibi, and M. F. Frias, ‘‘Inference engine based on closure and join operators over truth table binary relations,’’ *J. Logical Algebraic Methods Program.*, vol. 83, no. 2, pp. 180–193, 2014.

[2] G. Töpper, M. Knuth, and H. Sack, ‘‘Dbpedia ontology enrichment for inconsistency detection,’’ in *Proc. 8th Int. Conf. Semantic Syst.*, 2012, pp. 33–40.

[3] N. Hollenstein, N. Schneider, and B. Webber, ‘‘Inconsistency detection in semantic annotation,’’ in *Proc. LREC*, Portoro, Slovenia, 2016, pp. 3986–3990.

[4] F. Gutiererz, D. Dou, S. Fickas, and G. Griffiths, ‘‘Online reasoning for ontology-based error detection in text,’’ in *OTM Confederated International Conferences On the Move to Meaningful Internet Systems*. Springer, 2014, pp. 562–579.

[5] F. Bannay, M.-C. Lagasquie-Schiex, W. Raynaut, and P. Saint-Dizier, ‘‘Using a SMT solver for risk analysis: Detecting logical mistakes in texts,’’ in *Proc. IEEE 26th Int. Conf. Tools Artificial Intell.*, Nov. 2014, pp. 867–874.

[6] S. H. Tan, D. Marinov, L. Tan, and G. T. Leavens, ‘‘@Comment: Testing javadoc comments to detect comment-code inconsistencies,’’ in *Proc. IEEE 5th Int. Conf. Softw. Test.*, Apr. 2012, pp. 260–269.

[7] S. Motte, R. Ruti, A. Jacolin, and P.-A. Ruquier, ‘‘Apparatus, systems and methods for scoring and distributing the reliablity of online information,’’ U.S. Patent 20 160 239 496 A1, Sep. 24, 2014,

[8] H. Van Der Aa, H. Leopold, and H. A. Reijers, ‘‘Comparing textual descriptions to process models—The automatic detection of inconsistencies,’’ *Inf. Syst.*, vol. 64, pp. 447–460, Mar. 2017.

[9] B. Carmeli, R. Rinott, and N. Slonim, ‘‘Inconsistency detection between structured and non-structured data,’’ U.S. Patent 20 140 214 732, A1, Jan. 28, 2013.

[10] A. Pariyar, Y. Murakami, D. Lin, and T. Ishida, ‘‘Inconsistency detection in multilingual knowledge sharing,’’ *J. Inf. Knowl. Manage.*, vol. 13, no. 4, p. 1450033, 2014.

[11] H. Suleman, ‘‘Developing web metrics for measuring the quality of islamic websites,’’ M.S. thesis, Dept. Kulliyyah Inf. Commun. Technol., Int. Islamic Univ. Malaysia, Kuala Lumpur, Malaysia, 2005.

[12] J. Al Otaibi, S. Elloumi, A. Jaoua, and A. Hassaine, ‘‘Using conceptual reasoning for inconsistencies detection in islamic advisory opinion (fatwas),’’ in *Proc. IEEE/ACS 12th Int. Conf. Comput. Syst. Appl. (AICCSA)*, Nov. 2015, pp. 1–5.

[13] B. Ganter and R. Godin, *Formal Concept Analysis*. Berlin, Germany: Springer, 2005.

[14] S. Elloumi, A. Jaoua, B. Boulifa, M. Saleh, and J. A. Otaibi, ‘‘Conprove: A conceptual prover system,’’ in *Proc. IEEE/ACS 11th Int. Conf. Comput. Syst. Appl. (AICCSA)*, Nov. 2014, pp. 647–651.

[15] A. Hassaine, S. Elloumi, F. Ferjani, and A. Jaoua, ‘‘Hyper rectangular trend analysis application to islamic rulings (fatwas),’’ in *Proc. IEEE/ACS 11th Int. Conf. Comput. Syst. Appl. (AICCSA)*, Nov. 2014, pp. 318–325.

[16] A. Jaoua, ‘‘Pseudo-conceptual text and web Structuring,’’ in *Proc. 16th Int. Conf. Conceptual Struct. (ICCS)*, 2008, pp. 22–32.

[17] A. Hassaine, S. Mecheter, and A. Jaoua, ‘‘Text categorization using hyper rectangular keyword extraction: Application to news articles classification,’’ in *Proc. Int. Conf. Relational Algebraic Methods Comput. Sci.*, 2015, pp. 312–325.

[18] F. Ferjani, A. Jaoua, S. Elloumi, and S. B. Yahia, ‘‘Hyper-rectangular relation decomposition and dimensionality reduction,’’ in *13th Int. Conf. Relational Algebraic Methods Comput. Sci. (RAMiCS)*, 2013. [Online]. Available: http://www.cl.cam.ac.uk/events/ramics13/Ferjani_RAMICS13-01082012_updated.pdf

[19] E. F. Codd, ‘‘A relational model of data for large shared data banks,’’ *Commun. ACM*, vol. 13, no. 6, pp. 377–387, 1970.

[20] A. Liaw and M. Wiener, ‘‘Classification and regression by randomforest,’’ *R News*, vol. 2, no. 3, pp. 18–22, 2002.

[21] T. K. Ho, "Random decision forests," in *Proc. 3rd Int. Conf. Document Anal. Recognit.*, vol. 1. Aug. 1995, pp. 278–282.

JAMEELA AL OTAIBI received the B.Sc. degree in computer science from Qatar University, in 2003, the master's degree in computer science from Liverpool University in 2010. She is currently pursuing the Ph.D. degree in computer science with Qatar University. Her Ph.D. thesis entitled Information Extraction and Inconsistency Detection on a Given Ontology.

ZEINEB SAFI received the B.Sc. degree in computing from Qatar University in 2015, where she is currently pursuing the master's degree in computing. She is currently a Research Assistant, Department of Computer Science and Engineering. Her research interests include algorithm's design and analysis.

ABDELÂALI HASSAÏNE received the Ingenieur d'Etat degree in computer science from the Université de Tlemcen, Algeria, in 2005, the M.Res. degree in imaging and computer graphics from the INSA de Lyon, France, in 2006, and the Ph.D. degree in mathematical morphology from Mines ParisTech, France, in 2009. He is currently a Post-Doctoral Researcher with Qatar University. His research interests include image processing, data analysis, text mining, and machine learning.

FAHAD ISLAM received the bachelor's degree in computer science from Carnegie Mellon University, Qatar, in 2014. He is currently a Research Assistant in the scope of a granted project about Hidden Data Analysis. He developed different software on data reduction, knowledge extraction, and interfacing online application for small devices.

ALI JAOUA received the B.E. degree in computer science from the ENSEEIHT of Toulouse in 1977, the Dr. Eng. degree from the Institute Polytechnic of Toulouse in 1979, and the Ph.D. degree in computer science from the University Paul Sabatier of Toulouse, France, in 1987. He is a Professor in Computer Science and Engineering Department, University of Qatar, since 2000. His current research interests include automatic information extraction from Internet, data reduction, text mining, information retrieval, knowledge engineering, meta-search engine development, alerting system, natural language processing for arabic and english, pattern generation and recognition, and sentiments analysis. He published about 50 papers in international journals, presented about 100 conferences, and contributed to several books. He has been an Invited Professor and Researcher in the universities, such as University Joseph Fournier, Grenoble, France, in 2004, University of Sherbrook, Canada, in 1989, Sophia-Antipolis (I3S), France, in 1991. He has been an Associate Professor of Computer Science with Université Laval, Quebec, Canada, 1989–1992. He is a member of ACM Society.