IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 24 June 2023, accepted 22 July 2023, date of publication 1 August 2023, date of current version 8 August 2023.
Digital Object Identifier 10.1109/ACCESS.2023.3300737

RESEARCH ARTICLE

ArEmotive Bridging the Gap: Automatic Ontology Augmentation Using Zero-Shot Classification for Fine-Grained Sentiment Analysis of Arabic Text

AMER JARADEH AND **MOHAMAD-BASSAM KURDY**
Department of Web Sciences, Syrian Virtual University, Damascus, Syria
Corresponding author: Amer Jaradeh (amer_144615@svuonline.org)

ABSTRACT Human-computer interaction remains one of the final frontiers to conquer while held in perspective with the rapid developments and technology growth over recent years. It is an arduous task to convey the true human intent to the machine in order to generate a computerized relevant decision in a certain field. In recent years, focus has shifted to cover fields of study that relate to Sentiment Analysis (SA) to improve and ease the tasks of our daily lives. We Propose ArEmotive (Arabic Emotive), a fine-grained sentiment analysis system that is human-independent which can automatically grow its source of information allowing for more precision and a greater dataset each time it is used through ontology augmentation and classification. Our proposed architecture relies on multiple data sources running through certain pipelines to generate a central online repository utilized by any mobile system to access this info-base. This system is important because many researchers in the field of automated ontology alignment and ontology mapping achieved a semi-automated approach to map new ontologies out of old ones or to extend already existing ontologies with data from new ones. ArEmotive identifies fine-grained emotions in text based on a dynamic ontology enriched through ontology alignment, mapping and machine learning assisted classification, resulting in a structure that contributes in: a centralized dataset ever growing to fit the need of the users, a sustainable structure able to allocate new data sources without the need to modify the system, ability to generate appropriate information even with the absence of ‘‘parent’’ sources.

INDEX TERMS Arabic NLP, fine-grained emotions, ontology augmentation.

I. INTRODUCTION

An expert system must have access to relevant data enabling it to generate a right decision [1]. The ontology augmentation mechanism proposed in this study can achieve an ever-growing suitable dataset through a manner of automated sentiment analysis and text classification without human intervention or guidance. Ontology Alignment (OA) [2] is the act of creating a bridge ontology from two or more ontologies by finding common grounds shared between source ontologies. Usually, OA is guided by some sort of human intervention whether it is by approving and rejecting the result (the new system-found bridge structure between ontologies) or modifying it. This process is a prominent technique to overcome the restrictions and specifications of information and knowledge when the studied domain has no clear boarders, or when it might grow without the need to modify the structure. Some researchers have reached a semi-automatic algorithm in OA depending on a supporting human decision [3]. These concepts of Ontology Alignment and Ontology Augmentation are adopted to reach a bridge ontology serving as shared data source that will keep expanding each time a new source ontology is added regardless of the structure and content of this newly added ontology. Moreover, there are vast amounts of data being created and shared by internet users around the world by the moment, all of which carry the initiator's sentiment and intent. This data might be in the form of videos, audios, images, or raw text. However,

The associate editor coordinating the review of this manuscript and approving it for publication was Arianna Dulizia.

text holds an advantage to those interested in SA as it is the most used conduit for expression on the Internet. Although Arabic is the official tongue of 22 countries around the world, it is relatively young and challenging in the field of natural language processing (NLP) and Emotion Detection (ED) [4]. With more than 360 million native speakers, Arabic language did not have its share of viable research compared to other, more technological languages like English or Chinese [5]. The proposed structure makes use of ontology alignment and augmentation techniques empowered by machine learning algorithms enables the system to make precise decisions and to gather data from multiple sources whatever may be the structure and the properties within. The proposed research was based on ''Emotive system'' [6] which is a semi-fine grained emotion detection system only for English language, the main motive is to apply fine-grained sentiment analysis on Arabic text (As there is not much literature for Arabic text) while trying to add value by leveraging classification and machine learning techniques with ontology techniques to achieve truly dynamic ontology.

A. RESEARCH QUESTIONS

Our proposed architecture was devoted to answer our main research questions.

- Can ontology alignment and ontology augmentation aid in the construction of a proper dynamic ontology ?
- Is an ontology based system able to extract fine-grained emotions from text supported by machine-learning techniques?
- Are multi-lingual pre-trained language models precise while dealing with Arabic text?

The rest of this paper is organized as follows: in Section II, we review related work in the literature, in Section III, we give a formal definition of ontologies, ontology alignment and augmentation before providing an overview of our framework ArEmotive and describing the strategies used in Section IV. Section V shows the limitations encountered by ArEmotive and we give the experimental results. In Section VI, we conclude the paper and explore what may be some of our future works.

II. RELATED WORK

This section exhibits a number of related previous studies as this paper adopts some of their approaches and overcomes the absence of some points and aspects in others: RiMOM [7] achieved formalization of the problem of dynamic multi-strategy selection in the ontology alignment and defined the major tasks in dealing with it. The authors defined two similarity factors which quantitatively estimate the similarity characteristics between two ontologies. Then proposed a comprehensive framework to dynamically select and combine individual ontology alignment strategies considering both the textual and structural similarity metrics of two ontologies. AUTOMS [8] introduced a tool for automatic alignment of domain ontologies that integrated between several matching methods to ensure high precision and recall with the minimum human involvement. Others have come up with a bridge ontology that handles the interoperability between the source ontologies that represent the same or complementary application domain through combining matching source ontologies [9]. Moreover Cassab and Kurdy [10] have devised an approach to classify emotion in Arabic text into fine-grained emotions and achieve high precision an accuracy based on ontologies. Another research that tackled the fine-grained emotion problem in Arabic text [11] took a multi-label multi-target approach Whereas the ''Emotive'' system [6] achieved Extracting Fine-grained Emotions from Terse, Informal Messages where an ontology based system was built for fast and efficient capturing of a wider and a more comprehensive range of human emotions as opposed to coarse-grained emotional systems. Their system focused on informal English language making use of multiple urban dictionaries. As for Antoniazzi and Viola [12] who went further into dynamic ontologies where they utilized built-in Web Ontology Language (OWL), Resource Description Framework (RDF), Resource Description Framework Schema (RDFS) relations and properties to infer new ones and to construct a dynamic ontology structure, one of many which were surveyed to focus on evolving ontologies by relying on external background knowledge sources, rather than on user input [13]. Kalibatiene and Vasilecas [14] surveyed most used languages to represent ontologies and presented an account of their advantages and disadvantages and how it can affect the efficiency of an ontology based system. With the advancement of big transformer-based language models, BERT has been used to perform sentiment analysis and emotion detection [15], [16]. He et al. [17] have proposed a novel, general ontology matching system (BERTMap) that exploits the textual information and the structure of an ontology to learn word semantics and contexts effectively by utilizing the contextual embedding model BERT. As this paper confronts two main subjects (fine-grained emotions and dynamic ontologies) we point that some of the formerly mentioned literature have treated emotions as a whole entity or a general polarity (positive - negative), and others have achieved semi-dynamic ontologies that require human intervention in their process. This motive and goal of this research is to reach a ''Fully Dynamic ontology'' not a semi-dynamic one, where human intervention is not needed. (as mentioned in the introduction and in the beginning of section IV) and also to treat emotions as fine-grained where a certain text can contain multiple complex emotions.

III. PROBLEM AND TERM DEFINITIONS

In this section we define the main terms used throughout the article and highlight the problems within the process of fine-grained sentiment analysis using dynamic ontologies.

A. FINE-GRAINED EMOTIONS

In our proposed system, sentiments in major emotional theories [18] proposed by Drummond, Plutchik, Eckman, Izard [19] have been included with some new add-ons to better suit a studied use-case. Table 1 shows the emotions suggested by these emotional theorists which have been used by us alongside the ones introduced within ArEmotive found suitable to the studied use case. Some literature has perceived emotions in text as a whole polarity [20] marking them as positive, negative or neutral. Others went as far as marking text with an emotional class within this polarity [10]. However, to achieve higher accuracy we must consider that human sentiments are complex as Lindquist and Barrett [21] state, and a single word might carry numerous sentiments depending on its way of use and place in a sentence, as it is not limited to individual emotions in words or polarity as a whole. Moreover, a word in a single sentence might suggest multiple emotions within [22]. Thus, a wide range of sentiments have been added to aid the goal of this research like (Resolve, Hope, Pride, Concern, Envy, etc…) ranging in scale from -10 (warningly negative) to +10 (highly positive).

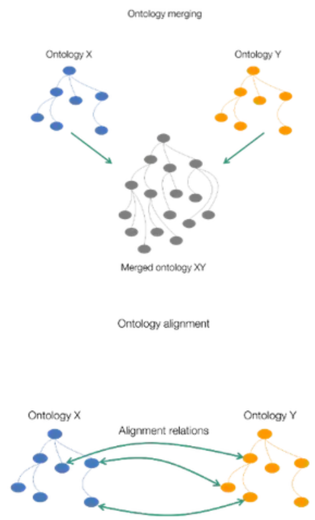

FIGURE 1. Illustration of ontology merging and alignment processes. [32].

B. ONTOLOGY

An ontology is a formal specification of a shared conceptualization [26]. Formally, an ontology may be represented as a tuple O $= < C, R, P, I, A >$, where C is a set of classes(or concepts of the domain), R is a set of relationships between the classes in C (also called object properties in some languages whose domain and range are classes in C), P is a set of data properties (a specific type of relation whose domain is a class and the range is a data type), I is a set of class instances (concrete objects of the classes) and A is a set of axioms [27].

C. ONTOLOGY MATCHING

Basically, ontology matching is a process in which links between entities of ontologies are established. Each semantic link is called a correspondence. And a set of correspondences is called an alignment [28]. Ontology matching identifies correspondences between the entities of multiple ontologies, and it is a necessary condition to establish interoperability between them to reach common grounds in order to achieve a desired goal [29].

D. ONTOLOGY ALIGNMENT

The result of a matching process "an alignment", is a set of pairs of entities E1 and E2 from two ontologies O1 and O2 that are supposed to satisfy a certain relation r with a certain confidence N. Given two ontologies O1 and O2, an alignment between O1 and O2 is a set of correspondences (i.e., 4-tuple): $\langle E1, E2, R, N\rangle$ with $E1 \in O1$ and $E2 \in O2$ being the two matched entities, R being a relationship holding between E1 and E2, and N expressing the level of confidence in this correspondence [30].The best way to compare results with confidence is to plot their precision/recall numbers. The examples are only provided for simple ontologies, which are class hierarchies but do not depend on this simple language as presented in [31]. The act of ontology alignment can be defined as search for a shared structure between two ontologies to find equal nodes (classes, instances) of one ontology with another as illustrated in Figure 1.

E. ONTOLOGY MAPPING

Ontology mapping [33] is described as finding the appropriate classes and relationships that correspond between two ontologies by Identifying Equal Attributes and calculating attribute similarity [34]. We understand ontology mapping as the task of relating the vocabulary of two ontologies sharing the same domain of discourse in a way that the mathematical structure of ontology signatures and their intended interpretations, as specified by the ontological axioms are respected [35].

F. ONTOLOGY AUGMENTATION

This field focuses on the addition process executed on a certain base ontology to add new instances, classes or relations in a manner that can be automated without human intervention or without having the need for a human factor to be present to guide the semi-automated process. This is a key aspect when considering dynamic ontologies [36].

G. ZERO-SHOT LANGUAGE MODELING

Zero-shot text classification aims to associate the appropriate label with a matching piece of text, irrespective of the domain of the text nor the aspect (e.g., topic, emotion, event, etc.) described by the label. Yin et al. [37] proposed a method for using pre-trained Natural Language Inference (NLI) models as a ready-made zero-shot sequence classifiers. The method works by posing the sequence to be classified as the NLI premise and to construct a hypothesis from each candidate label. In our proposed research "bart-large-mnli", "Roberta-large-xnli", "deberta-v3-base-mnli" [38], [39], [40] are utilized as three pre-trained (zero-shot) language models to infer zero-shot predictions from these models about a token's affiliation to a given emotional class (label), we rely on the

TABLE 1. Overview of supported emotions (aka. classes) by major emotional theories vs. ArEmotive. ArEmotive supports all the listed emotions across these theories.

Drummond [23]	*Ekman [24]*	*Izard [25]*	*Plutchik [19]*	*ArEmotive*
Anger	Anger	Anger	Anger	Resolve
Caring	Disgust	Contempt	Anticipation	Hope
Depression	Fear	Disgust	Disgust	Pride
Fear	Happiness	Fear	Fear	Depression
Happiness	Sadness	Guilt	Joy	Concern
Hurt	Surprise	Interest	Sadness	Optimism
Inadequateness		Joy	Surprise	Envy
Loneliness		Sadness	Trust	Relief
Remorse		Shame		
		Surprise		

dedicated zero-shot classification pipeline from the Hugging Face Transformers library, [41]. This zero-shot classification comes into play in two stages in ArEmotive's structure, it is used while mapping properties from base ontology and other supporting ontologies (cf. Figure 6), it is also used for predicting a token's affiliation to a group of studied emotional classes as in Table 2.

IV. AREMOTIVE

Our proposed framework ArEmotive makes use of ontology augmentation, ontology alignment techniques and machine learning based classification to search for fine-grained emotions in Arabic text. It combines multiple methods of text similarity calculation, structure resemblance and zero-shot pre-trained classification to achieve a fine-grained emotional ontology which is dynamically augmented without human intervention through the process described in Figure 3 and the technical model illustrated in Figure 4.

In order to achieve a dynamic ontology able to sustain new instances from other ontologies(which will be referred to as the "supporting ontologies") that may vary in properties and structure, a dynamic structure was created that will allow for new input while preserving data characteristics. The properties of the aforementioned ontology are object properties and data properties in OWL language as illustrated in Figure2, these properties can be categorized into three main categories: **Node info**: spans the properties that catalogue human-readable information about individual instances.

- **label:** the human readable surface form of the token.
- **class:** pre-defined collection of types that a token could belong to.
- **emotion:** the information about a single emotion class found in an instance.

Emotion description: the meta data related to each emotion class that show its emotional weight in scale.

- **weight**: the emotional weight of a word depending on the emotional class found within scaling in range form -10 to +10. This weight comes from the two sources, it may come from a previously annotated dataset, and if it's not found in any supporting source of information, it is set according to a baseline set by the annotators for each emotional class. i.e. "love" was set by the annotators to be +7 in emotional weight.
- **dialect**: the dialect of the matched word in the source representing one of the Arabic dialects or classical Arabic.

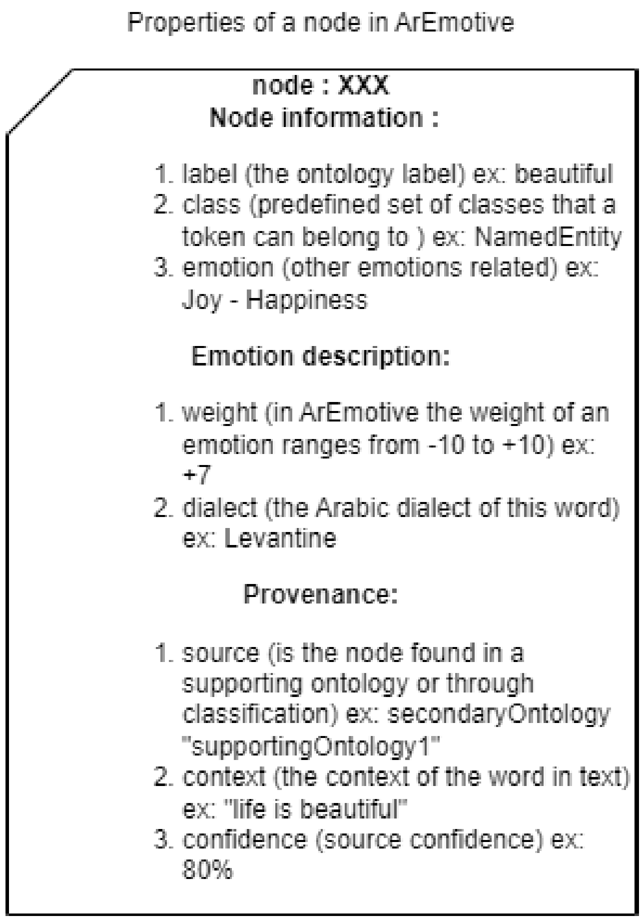

FIGURE 2. A detailed example of ArEmotives' node properties.

Provenance: the collection of properties that describe provenance information and meta data.

- **source:** The name of the source of the extracted emotion.
- **context:** Possible context of where the instance token could occur in natural language text.
- **confidence:** Numeric value of how the source was confident of its result.

An example of a property in ArEmotive ontology such as *"source"* property as a group of triplets:

```
<rdf:RDF xmlns:AE ="ArEmotive URI">
AE:source rdf:type owl:ObjectProperty
AE:Beautiful AE:source AE:supportingOntology1
```

After each life cycle, which means each time the system is run and a new modified ontology is generated, this ontology will act as a connection between various data sources. Its purpose is to enhance accuracy and reduce the time required to access information. These data sources consist of supporting ontologies developed by the authors. They are used to verify the system's functionality and imitate real-life ontologies that researchers can upload to the"Fuseki" server [42] connected to ArEmotive. This life cycle can be summarized in firstly, gathering input and search preferences from the user, then running this input through multiple NLP pipelines to generate tokens compatible with ArEmotive's structure, then searching for these tokens in the bridge ontology and the supporting ontologies, and returning the information if found or generating new tokens through the classification unit (Figure 8) if not found in any sources. The same classification algorithm is used to map the supporting ontologies' properties to our own. Which enables ArEmotive to know the exact property to search for while searching in a new ontology. The usage of the online server "Fuseki" to host ontologies enables access to them online and moves the processing logic away from the user's side to improve time costs and efficiency by using web APIs. This also is a key ingredient - along with the system's dynamic structure - to enable the users to add any new ontology they may access to the system's pool of ontologies to serve as a new source to search from thus increasing the accuracy each time it is used. This enabled the system to allocate for new sources like the supporting ontologies "SecondaryOntology", "DynamOnto", "testEmotive" which were initially seeded with 400, 350, 200 instances respectively. We gathered these instances by collecting different texts/tweets from the social media platform (Twitter). The growth of the bridge ontology then comes from two sources:

- The supporting ontologies(different parent sources) such as the mentioned "SecondaryOntology", "DynamOnto", "testEmotive".
- Classification (if not present in parent sources or supporting ontologies).

We maintained a difference in structure and properties between these ontologies within themselves and with the (base ontology) ArEmotive, which will play a key role in the system's dynamic structure. We handled this task under the supervision of three psychology students. The search process through Twitter was done by creating an application on Twitter's developer portal, then using this application and "Tweepy" to crawl for tweets through Twitter. Our process starts with raw data input by user to be queried to find relevant text through a social media sources (Twitter). This later on is run through multiple NLP pipelines to generate tokens relating to previous user input which can be handled on the ontology's side after it is converted to suitable tokens. We conduct an information gathering process for the newly found instances to create ontology instances with filtering and measuring adding up to the base ontology to match an instance's structure of ArEmotive ontology such as the one mentioned in Figure 5.

The mentioned process is controlled and directed by multiple units in the structure. These units are the user interface (UI) unit, the natural language processing (NLP) unit, the classification unit and the ontology handling unit. Each responsible for certain tasks in the ArEmotive flow, refer to Figure 3 for more details. This process follows the technical model described in Figure 4.

The scenario begins with the UI unit, which serves as the initial component. Its responsibility is to handle the query stage, which involves the user interface and a search engine for the social media platform Twitter. In this stage, text related to the user's queries is produced as output, which then becomes the input for the subsequent stage. Next in line is the natural language processing unit, which operates during the processing stage. Its purpose is to generate appropriate tokens that are suitable for classification and sentiment analysis. The tokens generated by the natural language processing unit are then passed to the measuring stage. In this stage, classification techniques, similarity calculation algorithms, and additional ontology queries are applied to these tokens. The outcome of this stage is the creation of new ontology instances, which serve as input for the augmentation stage. We accumulate all the needed information (emotional information) about a token to match ArEmotive's ontology structure including *emotional confidence*, *emotional weight*, *partial weight* which either come from a match in a supporting ontology or -if not found- a pre-defined default value defined by the annotators is set (see instance structure further in Figure5. The ontology handling unit receives newly found instances and updates the bridge ontology for it to be accessible in new queries. The ontology handeling unit relies on SPARQL queries with the "Fuseki" server [43] to perform DDL - DML on the hosted ontologies. Our running example that follows the process illustrated in Figures 3 and 4 starts with the user searching for a word or a sentence like ("AL HELWE" - which roughly means "the beautiful one" in the Levantine dialect) to end with an ontology instance that describes found emotional classes in this word depending on its context following the algorithm described in Algorithms 1 and 2.

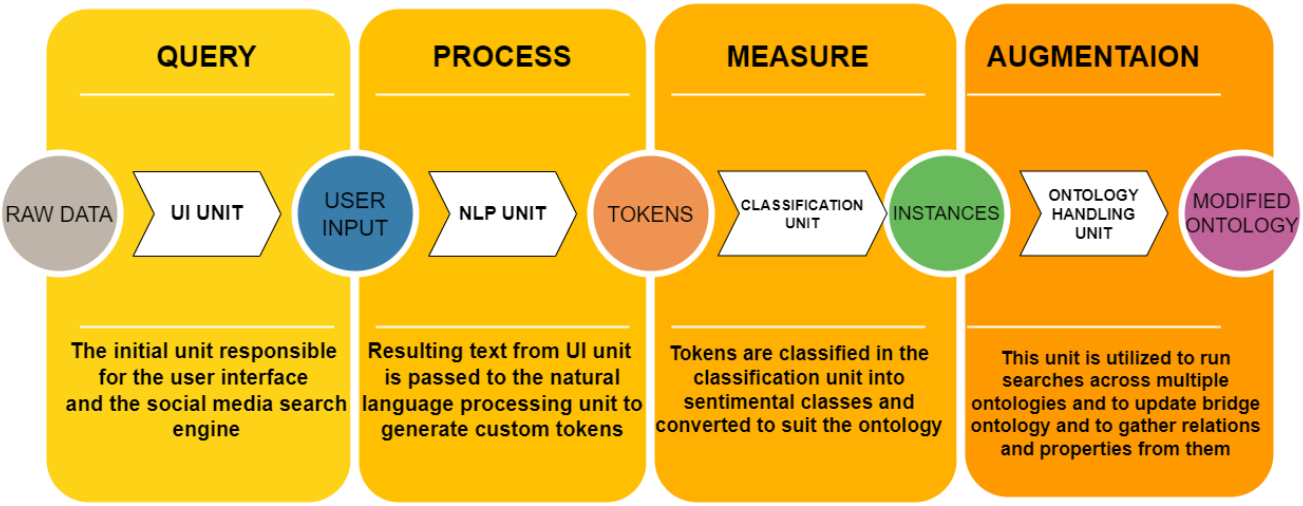

FIGURE 3. ArEmotive workflow expanded: overview of the phases with the corresponding logical processing units.

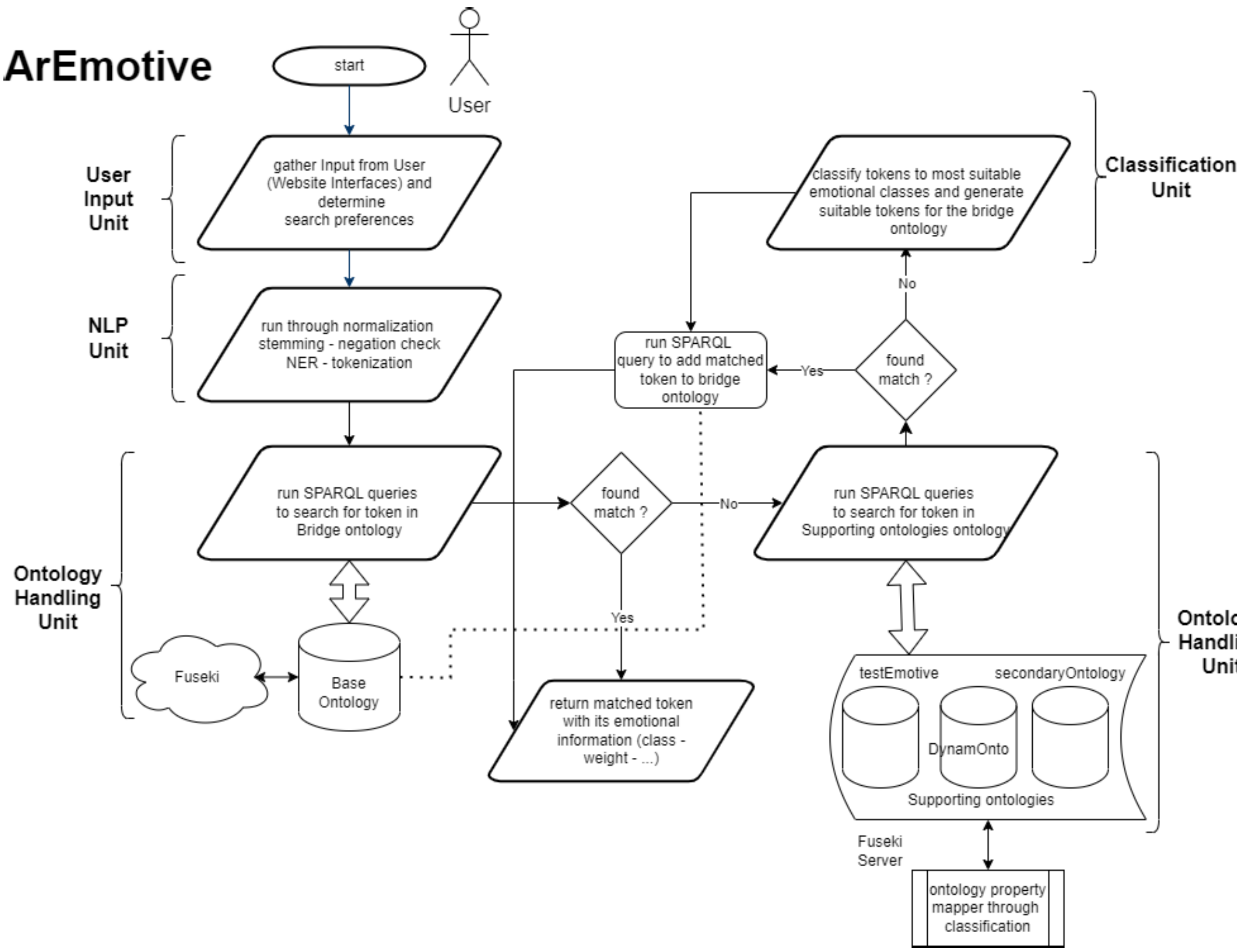

FIGURE 4. ArEmotive technical model: overview of the technical process followed in the proposed architecture.

This process relies on four units to achieve the fine-grained sentiment analysis task. The user interface unit accepts input from the user (''AL HELWE'') and parameters relating to the search choosing how many emotional classes to look for and

Algorithm 1 ArEmotive Main Algorithm

Input: text
Input: B (base ontology)
Input: S (supporting ontology)
Input: C (emotional classes)

text ← get input from user
tokens ← tokenize(text)
for *t* ∈ *tokens* **do**
    if *t* ∈ *B* **then**
        info ← B[t]
    else if *t* ∈ *S* **then**
        mappings ← classify(B, S)
        info ← S[t]
        info ← info + S[mappings for t]
        B ← B + info
    else
        info ← classify(t, C)
        info ← info + gatherDefaultInfo(C)[1]
        B ← B + info

1:gathering default information about a token and its relevant emotional classes to construct an ontology instance that corresponds with ArEmotive's ontology structure.

Algorithm 2 Classify function

Input: t (text)
Input: C (classes)
Output: results (token's affiliation to classes)

results ← []
for *m* ∈ *models* **do**
    for *c* ∈ *C* **do**
        classification ← transformer(t,c,m)
        results ← results + classification

return *avg(results)*

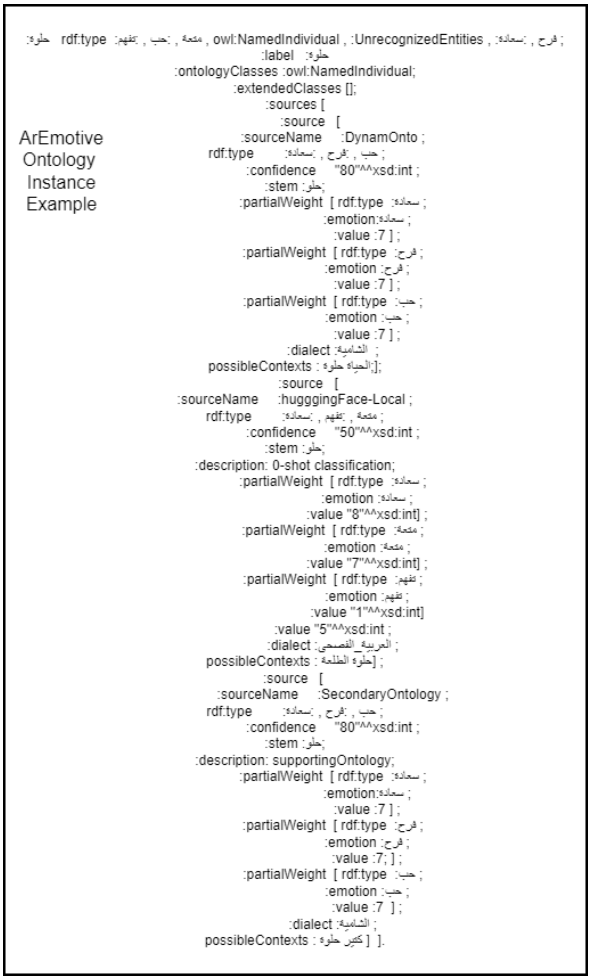

FIGURE 5. Serialized RDF representation of a single ArEmotive ontology instance.

the source to be searched (from a file, Twitter, exact search[2]). This search - depending on the search parameters - returns user created comments/tweets relating to the searched query through the social media platforms' (Twitter in our case) APIs like tweepy [44]. This results in a collection of raw unprocessed data that is passed on to the next stage as input. We process found text from the query stage in the natural language processing unit and generate suitable tokens. Our system uses multiple custom consecutive NLP pipelines to achieve this task. These NLP pipelines are originally based on CAMEL-TOOLS[3] [45] and customized to fit the project's needs to include (stemming - normalization - NER - stop word removal - negation checking - tokenization). We start the NLP pipeline by cleaning text to remove identifiers, connectors and Diacritiziation. Afterwards, a tagging process starts to disambiguate words as possible by tagging them with a part of speech and with a negation operator, affirmation operator or a linking word. So "AL HELWE" becomes "HELW". After the tokenization process, relevant tokens are generated and stored along with some key information like named entity recognition of the text. {"token":"HELW","NE":0,"negation":0,"composed":0}

We then search for newly generated tokens in our base ontology checking for the literal token or variations of it (stems and different contexts). In case this token already exists in the base ontology,its details are returned to the user. Presuming it was not found, an ontology search takes place on all uploaded ontologies on the ontology server "Fuseki" (in our case three new ontologies were added "SecondaryOntology", "DynamOnto", "testEmotive"[4]). Initially we scan the

[2]exact search: we take the user's input literally and process it as the source of information whereas in other sources like Twitter we gather related text about the input from these sources to serve as our base text. i.e. to apply ArEmotive techniques to a file, exact search is suitable

[3]CAMEL-TOOLS is a NLP and sentiment analysis library for Arabic language, where we utilized its NLP pipeline as a baseline to our own.

[4]SecondaryOntology, DynamOnto, testEmotive are the supporting ontologies used by ArEmotive to extend its base ontology. We created these three ontologies to test ArEmotive's ability to scan other ontologies and gather information from them while differing in structure and properties.

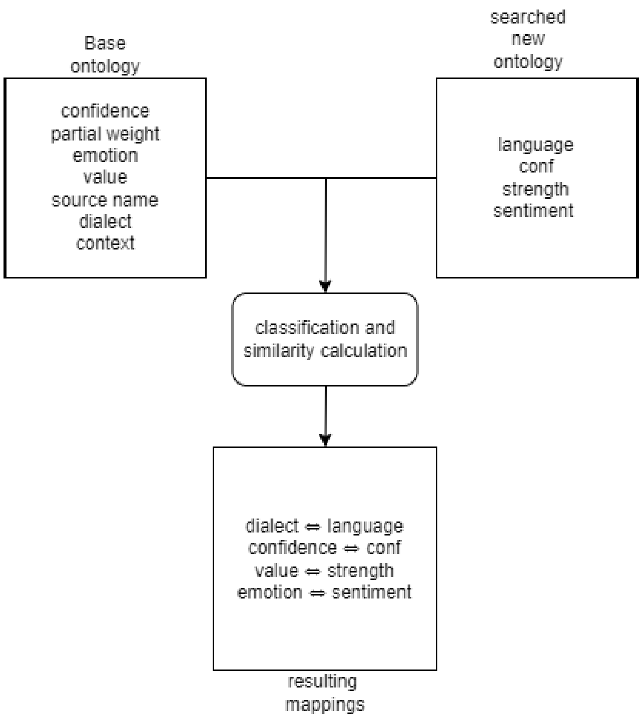

FIGURE 6. Example of a property mapping process between two ontologies to reach an agreement between each property and its mirror.

ontology for instances (objects) and relations (properties) to look for the queried term through SPARQL queries. If found, mappings are created between our base ontology properties and the searched ontology's using zero-shot classifiers. These mappings are created by calculating syntactic and semantic similarity between the scanned properties of the new ontology and those in our base ontology. We classify each of the new properties to classes that are the base ontology's properties to reach a correct decision on what properties in supporting ontologies mirror in the base ontology cf. Figure 6. After mapping the properties, we are now able to know what the information in other ontologies represent corresponding with our base ontology i.e., we now know that the relation *confidence* in "ArEmotive" ontology should have the information that is listed in "other ontology" under relation *emotional confidence*. This data is now fit to our needs and can be inserted into the base ontology with an initial source confidence that will be used in later stages to improve precision. see Figure 7 for more details.

In the case in which we do not find the queried terms in our base ontology nor our additional ontologies, multiple classification routes are pursued to reach our goal with fine-grained sentiments. An initial sentiment analysis is conducted to provide a whole emotional polarity (positive - negative) ("AL HELWE" is positive). Another -more fine-grained - classification happens between the studied word and the emotional classes considered in our project (34 emotional classes). This classification is done by utilizing three pre-trained models (bart-large-mnli, Roberta-large-xnli, deberta-v3-base-mnli) through transformers to calculate the word's affiliation to each of the mentioned emotional classes. These models were trained on the MultiNLI, Fever-NLI and Adversarial-NLI (ANLI) datasets [46], which comprise 763,913 NLI hypothesis-premise pairs and fine-tuned on XNLI, which is a multilingual NLI dataset [47].Noting that we used multilingual models to allow the classification of Arabic text. If the word's affiliation with the emotion class crosses a pre-defined threshold of precision (85%) (experimentally found to be suitable), the predictions are considered valid and added as emotional classes to the instance. This classification process relies on the three mentioned pre-trained language models to classify the text tokens to the emotional classes, where the tokens and the classes are passed to each model to calculate each token's affiliation to a given class. Afterwords, we average the three affiliations (from each model) to reach an intermediate value. The biggest values indicate the most affiliated classes with the token as illustrated in Figure 8.

TABLE 2. An example of ArEmotive's zero-shot classification results on the running example of "Al Helwe" classified with the emotional classes included in ArEmotive ranging between 0 and 1.

Word/Token	Emotional Class	Calculated Affiliation
Al Helwe	Anger	0.47825
Al Helwe	Caring	0.96211
Al Helwe	Depression	0.32196
Al Helwe	Fear	0.33907
Al Helwe	Happiness	0.88602
Al Helwe	Hurt	0.21456
Al Helwe	Inadequateness	0.00212
Al Helwe	Loneliness	0.12472
Al Helwe	Remorse	0.20741
Al Helwe	Disgust	0.19963
Al Helwe	Contempt	0.45536
Al Helwe	Sadness	0.36225
Al Helwe	Surprise	0.44134
Al Helwe	Joy	0.96621
Al Helwe	Shame	0.26839
Al Helwe	Interest	0.62395
Al Helwe	Trust	0.44288
Al Helwe	Relief	0.50607
Al Helwe	Optimism	0.55812

The classifications mentioned in Table 2 are calculated by each of models used in ArEmotive ('bart-large-mnli'', "Roberta-large-xnli", "deberta-v3-base-mnli"). Each model calculates the classification of a token to a class, and then we average the results of all models to end with a general classification that ArEmotive bases its decision upon. The instances generated by previous stages are now added to the base ontology to appear in future searches without the need to take up redundant time and effort searching for and classifying the tokens again. While querying the ontologies, if a match occurs, the process creates a sub-process that continues the search in other sources to update the previously mentioned *source confidence* to signify the percentage of credibility of the source. In the case of finding the token in other supporting ontologies, we check the emotional classes related to it in these sources. If a match is found within our base ontology, the source confidence is increased by a factor that is determined depending on the initial source's confidence and number of matches of emotional classes within the token.

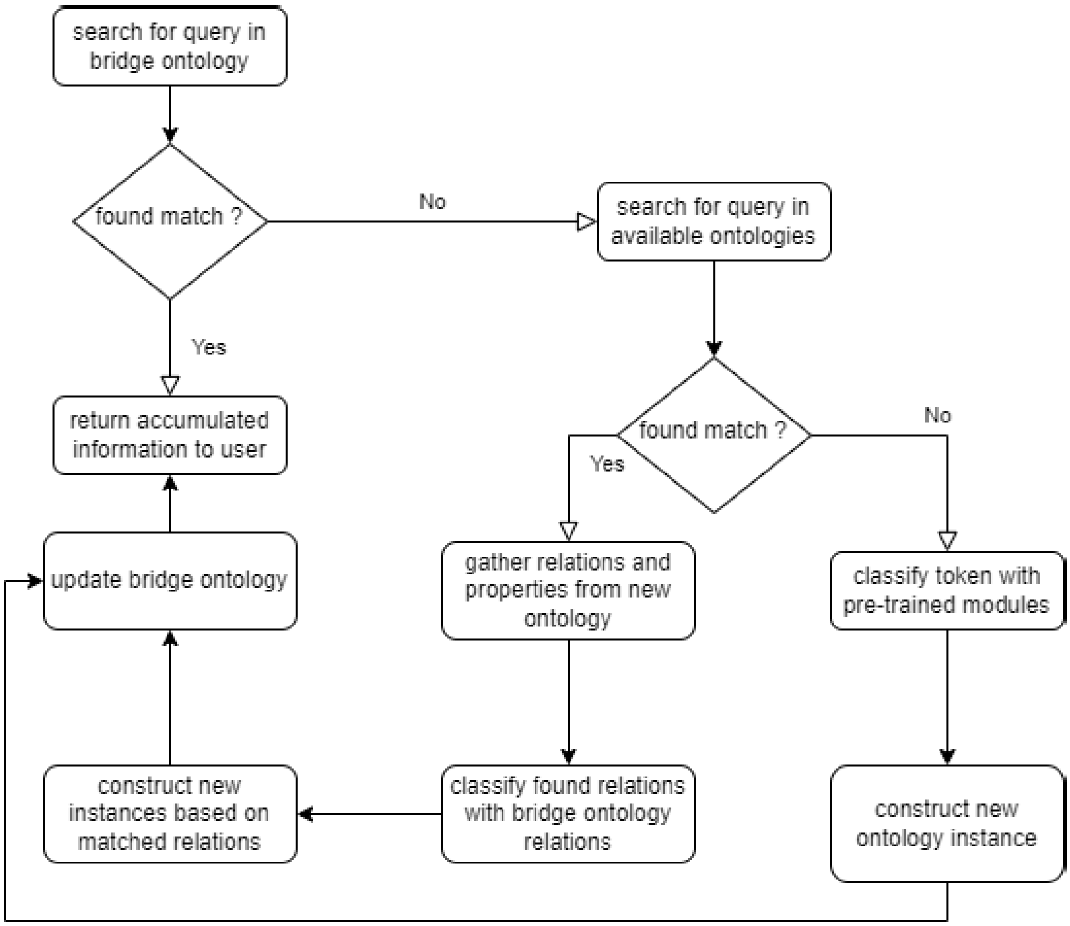

FIGURE 7. Detailed step-by-step procedure of the ontology handling unit.

However, if the token is found in other sources with a different emotional classes from the ones in the base ontology, the confidence of the source decrease by the same factor. This factor plays a critical role in deciding what to show the end user and how it is presented. Our running example shapes up to be as illustrated in Figure 9. The base ontology initially did not have any instances. after testing this ontology grew to over 3700 instances, indicating the ability of growth in the proposed structure.

V. LIMITATIONS, RESULTS AND DISCUSSION

In this section we exhibit the limitations and results of ArEmotive comparing it to others and we perform multiple measurement metrics on the data and the ontology.

A. LIMITATIONS

Our proposed system does not come up empty when searching for a new term in the base ontology because eventually it reaches a point -if it does not find a match- where it generates the required information by the use of pre-trained language models. However, this impacts performance. The precision of the emotional classes may decline while generating new instances with new emotion classes. The lack of Arabic emotion datasets -especially in the Levantine dialect- forced us to utilize zero-shot classifiers which impacted on the overall evaluation metrics negatively. Another limitation relating to the property mapping process between ontologies takes place in the case that a supporting ontology has many properties that are too similar, confusing the system to make a wrong decision while mapping properties. Thus, updating instances with information from a wrong property.

B. ONTOLOGY SIZE

The dynamic structure of our proposed system makes our base ontology a dynamic one, not a static source of date but growing and changing. This means that at the beginning of our testing phase, the base ontology was not seeded (0 instances). However, during the development process, testing and live phases, each run of the process resulted in adding new instances to the base ontology reaching up to (3700 instances roughly) at the time of writing this paper with more than 25000 triplets in the ontology i.e. a new run means a larger more precise ontology and dataset.

C. DATA ACCURACY

The information retrieved by the system may come from an external source like the supporting ontologies mentioned

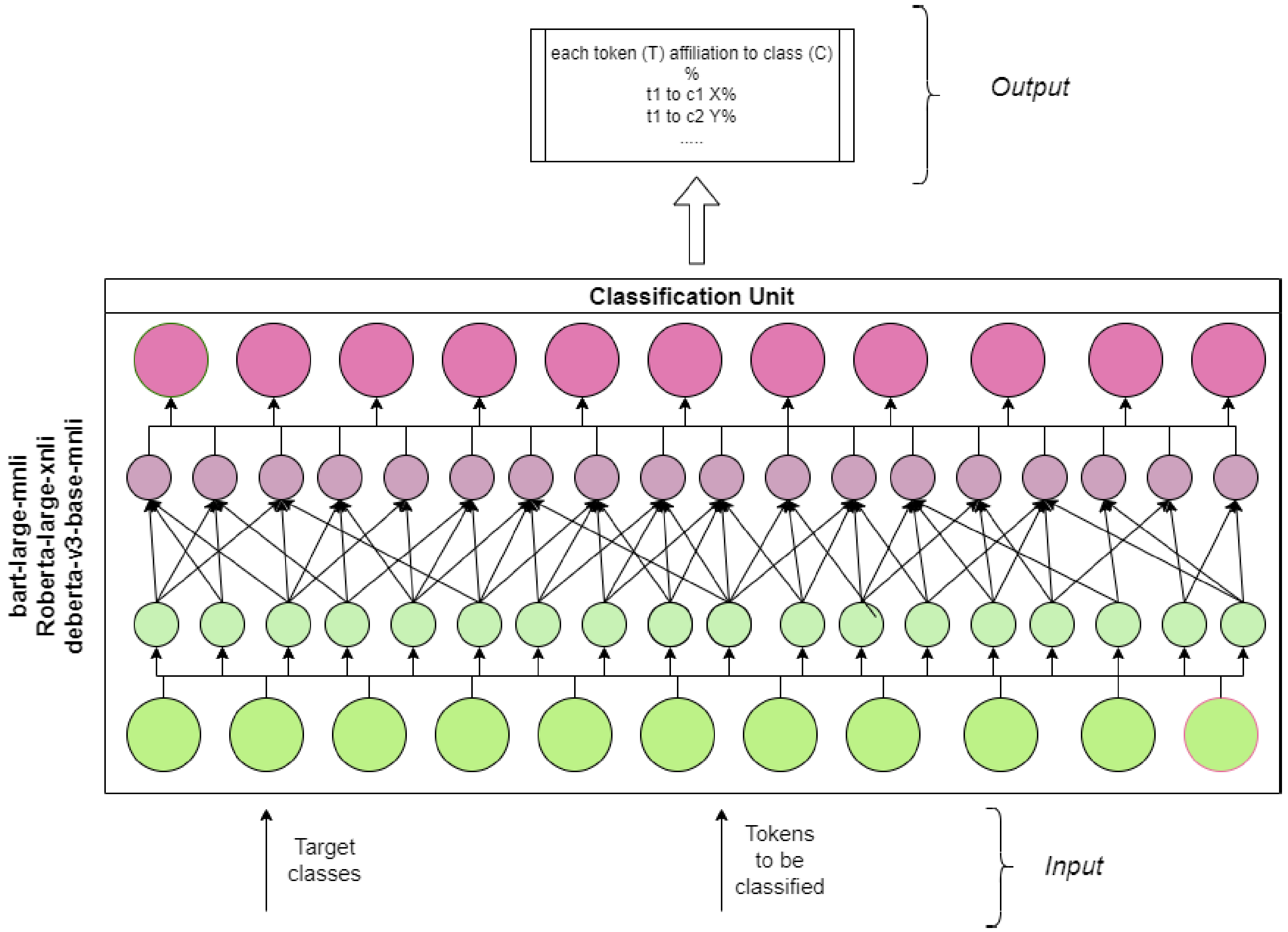

FIGURE 8. **Classification unit of ArEmotive in depth depicting the language models used in the process.**

earlier, so technically, there is no clear way to determine the accuracy of the returned information except for checking the recall of the fetched data and the precision of the generated data. To achieve this, we performed three experiments to evaluate the system's accuracy and to answer our research questions. Which brings us back to our research questions.

1) RESEARCH QUESTIONS ANSWERED

- Can ontology alignment and ontology augmentation aid in the construction of a proper dynamic ontology ? Yes, we have concluded that ontology alignment and ontology augmentation are key and the base for creating a dynamic ontology to relinquish any human intervention.
- Is an ontology based system able to extract fine-grained emotions from text supported by machine-learning techniques? Yes, an ontology-only based system is sometimes constrained to the data it holds. However, adding the ability to classify text or map between properties from different ontologies proved very helpful.
- Are multi-lingual pre-trained language models precise while dealing with Arabic text? The lack of Arabic language models was an issue, which drove us to use multi-lingual models, this certainly affected the precision of the results and a fine-tuned multi-lingual model or an Arabic model can truly boost the results and its precision.

2) EMOTION CLASSES ACCURACY

We used two styles of measurement to calculate the precision of generated data (i.e. when the system does not find any match in any source and predicts the fine-grained emotions in the text). A term known to be absent in the base ontology and present and the supporting ontologies is queried but removed from those supporting ontologies before the search process. The system in this case does not find matches and predicts sentimental classes. Classes that are matched with those removed from the supporting ontology to evaluate the precision of the data by comparing the emotional classes present in the term and other relevant properties between both ontologies. Another measurement takes place to calculate the accuracy of generated emotional classes i.e., should these classes come up as a result or should others. For this task, the help of three post-graduate psychology students was asked to individually classify an initial dataset that was used for the measurement process. In case of inter-annotator disagreement, a consensus mechanism is used to decide on which

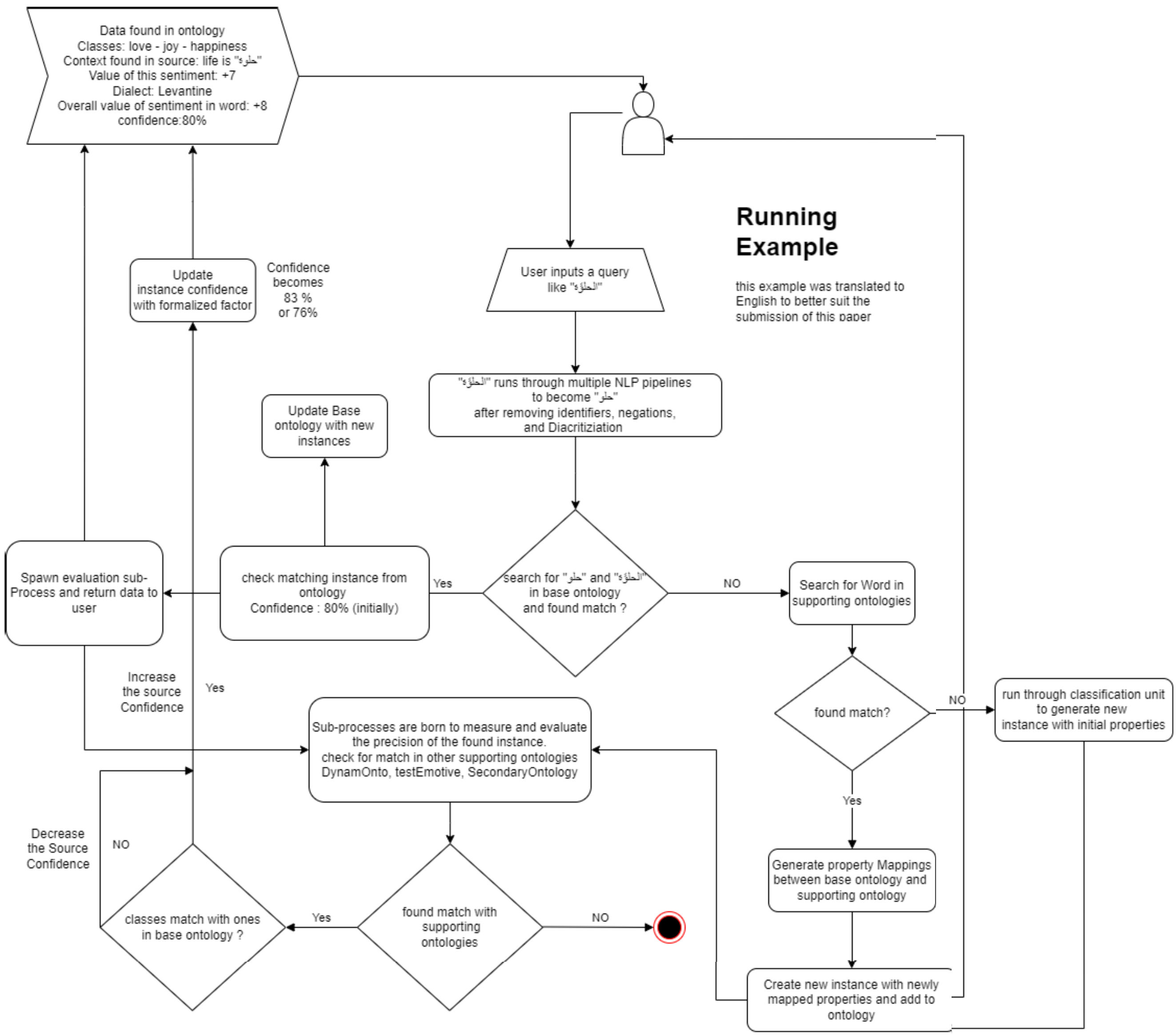

FIGURE 9. Running example of ArEmotive flow, depicting the progression of the example throughout an end-to-end run of the system.

annotation to use in the dataset. A consensus constitutes two out of three annotators agreeing on an annotation. We compared the emotional classes generated by our system with the classes annotated by the students. In our proposed system, we define our evaluation metrics to be precision, recall and F1 measure [48]. Since any given result in the evaluation dataset does not only contain a singular value rather multiple values, we leverage the @k variation of those metrics. In our experimental settings we set k = 10 where we found this number to be most suitable overall as we tried a range for @k from 1 to 15 with increments of 5 and found that the best results performance wise were for k = 10.

Table 3 shows the results of this evaluation in addition to the results of the ablation tests. In the ablation tests we study the effects of adding n-gram matching (scanning n-pairs of tokens) alongside exact matching (scanning the exact surface form of the token) and stem matching (scanning variations of the token). In case a match occurs in the search phase i,e., the system finds a match for a queried term in some supporting ontology, the same evaluation that took place while finding a match happens to determine the accuracy of returned emotional classes and their properties. This evaluation process is

TABLE 3. Ablation analysis results of ArEmotive in terms of precision, recall, and F1-measure.

	Precision	Recall	F1 measure
ArEmotive	73.3%	100%	84.59%
ArEmotive + n-gram	82%	100%	90.1%
ArEmotive + exact matching	71.6%	100%	83.44%
ArEmotive + stem matching	77.9%	100%	87.57%

TABLE 4. Evaluation results of property mapping in terms of different experimental setups (i.e. ontologies). The numbers in the table reflect the accuracy metric.

	SecondaryOntology	DynamOnto	testEmotive
ArEmotive	100%	80%	100%

done to measure whether these classes are the ones supposed to come up or not.

3) PROPERTY ACCURACY

While scanning a supporting ontology, we gather all relations and properties (data properties, and object properties) of this ontology. Another measurement process occurs to determine if all properties are found, also, if mapped properties between our base ontology and the supporting ontology are correct. In ArEmotive, we uploaded three supporting ontologies (SecondaryOntology, DynamOnto, and testEmotive) knowing what relations they have and what these relations describe. For the property extraction part, the system achieves a 100% recall in retrieving an ontology's properties, as this feature depends on basic OWL structure. Moreover, when the system does not find a its desired values it generates a new one, thus achieving 100% recall. Knowing the relations present in the supporting ontologies and what they represent enabled us to conduct an evaluation of the mapped properties i.e., Are the mapped properties between two ontologies correct? or should it be linked to anther property? This precision evaluation however came up with an overall precision of 90% including cases of multi-lingual restrictions as described in Table 4.

4) COMPUTATIONAL RESOURCES

ArEmotive (as a back-end or server not the user) does not noticeable resources to run, standard personal equipment were sufficient, an increase/upgrade of these resources will lead to improvements in the performance and time. Also, computation resource comparison was done on ontology storing formats (XML - N3 - NT) and found that NT format was best performance wise.

VI. CONCLUSION

In ArEmotive, we managed to achieve a sustainable dynamic ontology fit to handle fine-grained sentiment analysis tasks. This system considers emotions in text to be complex entities instead of positive/negative emotions. It takes into account the absence of the desired data from datasets and utilizes zero-shot classification to get its corresponding emotion classes. It also takes into consideration that supporting data sources (other ontologies) may vary in structure and properties, by using semantic/syntactic similarity calculation methods and classification techniques to map new-to-old properties allowing for an alignment to occur. Our next steps include 1) using ensemble learning to add fully supervised models to enable autonomous learning. 2) expansion to multi-lingual tools to transfer knowledge from non-arabic datasets and models to our own. 3) enable hyper-parameter tuning for social events or public response use-case tailoring such as micro news detection, and suicide prediction. 4) extend our supporting ontology pool with major ontologies like "wordNet".

DATA AVAILABILITY

The base ontology (ArEmotive) with the supporting ontologies are available at https://zenodo.org/record/7391143 alongside the source code of the framework that can be found at github repository: https://github.com/mosqit000/ArEmotive

CONFLICTS OF INTEREST

The authors declare that there is no conflict of interest regarding the publication of this paper.

FUNDING STATEMENT

No funding body.

ACKNOWLEDGMENT

The authors thank the three researchers that helped them with the annotation of their experimental data.

REFERENCES

[1] J. S. Edwards, Y. Duan, and P. C. Robins, "An analysis of expert systems for business decision making at different levels and in different roles," *Eur. J. Inf. Syst.*, vol. 9, no. 1, pp. 36–46, Mar. 2000.

[2] P. Pittet, C. Nicolle, and C. Cruz, "Guidelines for a dynamic ontology—Integrating tools of evolution and versioning in ontology," 2012, *arXiv:1208.1750*.

[3] J. Murdock, C. Buckner, and C. Allen, "Evaluating dynamic ontologies," in *Proc. Int. Joint Conf. Knowl. Discovery, Knowl. Eng., Knowl. Manage.* Cham, Switzerland: Springer, 2010, pp. 258–275.

[4] K. Shaalan, S. Siddiqui, M. Alkhatib, and A. A. Monem, "Challenges in Arabic natural language processing," in *Computational Linguistics, Speech and Image Processing for Arabic Language*. Singapore: World Scientific, 2019, pp. 59–83.

[5] A. Farghaly and K. Shaalan, "Arabic natural language processing: Challenges and solutions," *ACM Trans. Asian Lang. Inf. Process.*, vol. 8, no. 4, pp. 1–22, Dec. 2009.

[6] M. Sykora, T. Jackson, A. O'Brien, and S. Elayan, "Emotive ontology: Extracting fine-grained emotions from terse, informal messages," Loughborough Publications, London, U.K., Tech. Rep., 2013.

[7] J. Li, J. Tang, Y. Li, and Q. Luo, "RiMOM: A dynamic multistrategy ontology alignment framework," *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 8, pp. 1218–1232, Aug. 2009.

[8] K. Kotis, A. G. Valarakos, and G. A. Vouros, "AUTOMS: Automated ontology mapping through synthesis of methods," in *Proc. Int. Semantic Web Conf.*, Georgia, USA, 2006, p. 96.

[9] S. Amrouch and S. Mostefai, "Semantic integration for automatic ontology mapping," Comput. Sci. & Inf. Technol. (CS & IT), AIRCC, Chennai, India, Tech. Rep., 2013, pp. 387–396.

[10] S. Cassab and M.-B. Kurdy, "Ontology-based emotion detection in Arabic social media," *Int. J. Eng. Res. Technol.*, vol. 9, no. 8, pp. 1991–2013, 2020.

[11] O. Badarneh, M. Al-Ayyoub, N. Alhindawi, L. A. Tawalbeh, and Y. Jararweh, "Fine-grained emotion analysis of Arabic tweets: A multi-target multi-label approach," in *Proc. IEEE 12th Int. Conf. Semantic Comput. (ICSC)*, Jan. 2018, pp. 340–345.

[12] F. Antoniazzi and F. Viola, "Building the semantic web of things through a dynamic ontology," *IEEE Internet Things J.*, vol. 6, no. 6, pp. 10560–10579, Dec. 2019.

[13] F. Zablith, "Dynamic ontology evolution," Open Univ., London, U.K., Tech. Rep., 2008.

[14] D. Kalibatiene and O. Vasilecas, "Survey on ontology languages," in *Proc. Int. Conf. Bus. Informat. Res.* Cham, Switzerland: Springer, 2011, pp. 124–141.
[15] N. Alkaabi, N. Zaki, H. Ismail, and M. Khan, "Detecting emotions behind the screen," *AI*, vol. 3, no. 4, pp. 948–960, Nov. 2022.
[16] P. K. Jain, W. Quamer, V. Saravanan, and R. Pamula, "Employing bert-DCNN with sentic knowledge base for social media sentiment analysis," *J. Ambient Intell. Humanized Comput.*, vol. 2022, pp. 1–13, Jan. 2022.
[17] Y. He, J. Chen, D. Antonyrajah, and I. Horrocks, "BERTMap: A bert-based ontology alignment system," in *Proc. AAAI Conf. Artif. Intell.*, vol. 36, 2022, pp. 5684–5691.
[18] K. T. Strongman, *The Psychology of Emotion: Theories of Emotion in Perspective*. Hoboken, NJ, USA: Wiley, 1996.
[19] R. Plutchik and H. Kellerman, *Theories of Emotion*, vol. 1. New York, NY, USA: Academic, 2013.
[20] P. Rodriguez, A. Ortigosa, and R. M. Carro, "Extracting emotions from texts in e-learning environments," in *Proc. 6th Int. Conf. Complex, Intell., Softw. Intensive Syst.*, Jul. 2012, pp. 887–892.
[21] K. A. Lindquist and L. F. Barrett, "Emotional complexity," in *Handbook of Emotions*. Boston College, 2008.
[22] R. C. Solomon and L. D. Stone, "On 'positive' and 'negative' emotions," *J. Theory Social Behav.*, vol. 32, no. 4, pp. 417–435, 2002.
[23] T. Drummond. (2004). *Vocabulary of Emotions*. [Online]. Available: https://tomdrummond.com/wp-content/uploads/2019/11/Emotion-Feelings.pdf
[24] P. Ekman, "All emotions are basic," in *The Nature of Emotion: Fundamental Questions*. New York, NY, USA: Wiley, 1994, pp. 15–19.
[25] C. E. Izard, "Basic emotions, relations among emotions, and emotion-cognition relations," *Psychol. Rev.*, vol. 99, no. 3, pp. 561–565, 1992.
[26] M. Li, X.-Y. Du, and S. Wang, "Learning ontology from relational database," in *Proc. Int. Conf. Mach. Learn. Cybern.*, 2005, pp. 3410–3415.
[27] T. Gruber, "Ontology," Stanford Univ., CA, USA, Tech. Rep., 2018.
[28] J. Euzenat, A. Ferrara, L. Hollink, A. Isaac, C. Joslyn, V. Malaisé, C. Meilicke, A. Nikolov, J. Pane, and M. Sabou, "Results of the ontology alignment evaluation initiative 2009," in *Proc. 4th ISWC Workshop Ontol. Matching*, 2010, pp. 126–173.
[29] J. Euzenat and P. Shvaiko, *Ontology Matching*, vol. 18. Cham, Switzerland: Springer, 2007.
[30] S. Oppl and C. Stary, "Alignment of multiple perspectives: Establishing common ground for triggering organizational change," in *Designing Digital Work*. Cham, Switzerland: Springer, 2019, pp. 133–178.
[31] M. Ehrig, *Ontology Alignment: Bridging the Semantic Gap*, vol. 4. Cham, Switzerland: Springer, 2006.
[32] T. Chungoora. (2020). *Ontology Mapping Infographic. Ontology Merging, Transformation*. Accessed: Nov. 27, 2022. [Online]. Available: https://tishchungoora.medium.com/ontology-mapping-infographic-621272b4d445
[33] Y. Kalfoglou and M. Schorlemmer, "Ontology mapping: The state of the art," *Knowl. Eng. Rev.*, vol. 18, no. 1, pp. 1–31, Jan. 2003.
[34] J. De Bruijn, M. Ehrig, C. Feier, F. Martín-Recuerda, F. Scharffe, and M. Weiten, "Ontology mediation, merging and aligning," in *Semantic Web Technologies*. New York, NY, USA: Wiley, 2006, pp. 95–113.
[35] N. F. Noy and M. Klein, "Ontology evolution: Not the same as schema evolution," *Knowl. Inf. Syst.*, vol. 6, no. 4, pp. 428–440, Jul. 2004.
[36] J. Heflin and J. Hendler, "Dynamic ontologies on the web," in *Proc. AAAI/IAAI*, 2000, pp. 443–449.
[37] W. Yin, J. Hay, and D. Roth, "Benchmarking zero-shot text classification: Datasets, evaluation and entailment approach," 2019, *arXiv:1909.00161*.
[38] Y. Xian, C. H. Lampert, B. Schiele, and Z. Akata, "Zero-shot learning—A comprehensive evaluation of the good, the bad and the ugly," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 9, pp. 2251–2265, Sep. 2019.
[39] A. Gera, A. Halfon, E. Shnarch, Y. Perlitz, L. Ein-Dor, and N. Slonim, "Zero-shot text classification with self-training," 2022, *arXiv:2210.17541*.
[40] A. Alcoforado, T. P. Ferraz, R. Gerber, E. Bustos, A. S. Oliveira, B. M. Veloso, F. L. Siqueira, and A. H. R. Costa, "ZeroBERTo: Leveraging zero-shot text classification by topic modeling," in *Proc. PROPOR*. Cham, Switzerland: Springer, 2022, pp. 125–136.
[41] T. Wolf et al., "HuggingFace's transformers: State-of-the-art natural language processing," 2019, *arXiv:1910.03771*.
[42] A. H. Dantas and N. Cordeiro, "Apache Jena," HP Labs, Bristol, U.K., Tech. Rep., 2019.
[43] E. Sirin and B. Parsia, "SPARQL-DL: SPARQL query for OWL-DL," in *Proc. OWLED*, vol. 258, 2007, pp. 1–12.
[44] J. Roesslein. (2009). *Tweepy Documentation*. [Online]. Available: http://tweepy.readthedocs.io/en/v3
[45] O. Obeid, N. Zalmout, S. Khalifa, D. Taji, M. Oudah, B. Alhafni, G. Inoue, F. Eryani, A. Erdmann, and N. Habash, "Camel tools: An open source Python toolkit for Arabic natural language processing," in *Proc. 12th Lang. Resour. Eval. Conf.*, 2020, pp. 7022–7032.
[46] A. Nighojkar and J. Licato, "Mutual implication as a measure of textual equivalence," in *Proc. Int. FLAIRS Conf.*, vol. 34, 2021, pp. 1–22.
[47] A. Conneau, G. Lample, R. Rinott, A. Williams, S. R. Bowman, H. Schwenk, and V. Stoyanov, "XNLI: Evaluating cross-lingual sentence representations," 2018, *arXiv:1809.05053*.
[48] D. Vrandečić, "Ontology evaluation," in *Handbook on Ontologies*. Cham, Switzerland: Springer, 2009, pp. 293–313.

AMER JARADEH was born in Damascus, Syria, in 1995. He received the B.S. degree from the Faculty of Information Technology, Damascus University, in 2019. He is currently pursuing the M.S. degree in web sciences with Syrian Virtual University. Since 2022, he has been a Software Developer in a telecommunications company involved with the services integration and business applications and the e-payment rapid growth in Syria.

MOHAMAD-BASSAM KURDY was born in Damascus, Syria, in July 1961. He received the master's degree in information systems engineering from INPG, France, in 1986, and the Ph.D. degree in mathematical morphology from Mines ParisTech, France, in 1990. He was with HIAST, Damascus, Syria, from 1991 to 2013, and held the position of the Head of computer sciences, from 1997 to 2003, the EC Project EUMEDIS-Medforist Country Manager for Syria, and a Professor with Syrian Virtual University, ESC Dijon, and ESC Rennes. He teaches advances data mining, big data, information retrieval, and content based image retrieval (cbIR), and supervised over many postgraduates.