Exploiting a Multilingual Semantic Machine Translation Architecture for Knowledge Representation of Patient Information for Covid-19

Laurette Marais[1] and Laurette Pretorius[2]

[1] Digital Audio and Visual Technologies, Next Generation Enterprises and Institutions, CSIR, Pretoria, South Africa
laurette.p@gmail.com
[2] College of Graduate Studies, University of South Africa, Pretoria, South Africa
laurette@acm.org

Abstract. AwezaMed Covid-19 is a multilingual speech-to-speech translation application for screening patients for Covid-19. It enables English-speaking health care providers (HCPs) to conduct screenings by asking questions to patients in all other official languages of South-Africa. It uses a multimodal computational grammar translation system to enable English speech and screen-based input, which can be translated to produce synthetic speech in the target languages. Grammatical Framework is used for the translation system, utilising a semantic interlingua. Because of this, each utterance translated by the application is represented by a semantic tree, which could be exploited for knowledge representation.
In this paper, we describe how the machine translation architecture designed for multilingual speech-to-speech translation can be adapted for knowledge representation consistent with existing knowledge representation formalisms, namely openEHR archetypes and RDF triples, which could be recorded seamlessly by HCPs during the screening.

Keywords: Grammatical Framework · openEHR · Linked Data · RDF

1 Introduction

South Africa is a multilingual society with 11 official languages. English serves as a lingua franca in many spheres and it is the language in which most health care providers (HCPs) are educated at tertiary level. However, large numbers of South Africans are not proficient in English, creating a language barrier to health care in many settings. At hospitals and clinics, it is often the case that other staff, including security guards and cleaners, are called upon to interpret in cases where the HCP and patient do not share a common language. This has a detrimental effect on efficient use of staff, patient privacy and the ability of HCPs to provide respectful care.

The Covid-19 pandemic has focused attention on the ability of health care systems to cope with enormous amounts of patients at once, as well as the need to record information that may aid in understanding and responding to dangerous viral outbreaks. Both these factors are severely affected if language barriers exist between HCPs and patients.

To enable communication for screening patients for Covid-19, a speech-to-speech mobile translation application was developed to translate spoken English utterances to spoken utterances in all other official languages of South Africa. AwezaMed Covid-19 uses automatic speech recognition (ASR), grammar-based machine translation (MT) and text-to-speech (TTS) to enable communication by an HCP to a patient. The application was adapted from a different version of AwezaMed geared towards maternal health. [10]

In this paper, we describe how AwezaMed Covid-19 can be adapted for knowledge representation within existing frameworks, which could be recorded seamlessly by HCPs during the screening. In Section 2 we contextualise our work. Various factors contributed to the choice to provide translations in only one direction, namely from the HCP to the patient, which leads directly to the ability to enable knowledge representation, and we discuss them in Section 3. In Section 4 we describe the multilingual translation system, and in Section 5, we describe our main contribution, namely an extension to the translation system that enables knowledge representation of patient information. We discuss the implications of this proof-of-concept implementation in Section 6, before providing concluding remarks.

2 Contextualisation

The amount of patient data, including Covid-19 information, is constantly increasing world wide and there is an urgent need to have a clear picture of the development and spread of the pandemic, also in developing countries such as South Africa. Indeed, the rapid acquisition, publication and interoperability of such data have a high priority. In the past three decades various standards, vocabularies and knowledge representations have been developed for this purpose, for example, ISO standards,[3] SNOMED-CT,[4] openEHR,[5] HL7,[6] and FHIR.[7] This resulted in patient information [13] encoded in a variety of formalisms (knowledge representations).

openEHR is a technology for creating and managing electronic health care records (EHRs), "consisting of open specifications, clinical models and software that can be used to create standards, and build information and interoperability solutions for healthcare".[8] An important aspect of how openEHR deals with

[3] https://www.iso.org/files/live/sites/isoorg/files/store/en/PUB100343.pdf
[4] https://www.snomed.org/snomed-ct/five-step-briefing
[5] https://www.openehr.org/
[6] https://www.hl7.org/
[7] http://hl7.org/fhir/
[8] https://www.openehr.org/about/what_is_openehr

knowledge is its two-level model [5], consisting of the reference model (RM), which defines models of information content, such as data types and data structures, and the archetype model (AM), in which more specific domain knowledge can be described, such as the results of a laboratory test. The RM is stable and implemented in software,[9] while the goal of the AM is to allow clinicians to develop formal models that reflect their domain knowledge [7]. The Clinical Knowledge Manager (CKM) is a platform for sharing and collaborating on such domain knowledge models in the form of *archetypes* and *templates*.

In recent months, an archetype for Covid-19 symptoms[10] was contributed, in the form of a specialisation of the existing archetype for symptoms. A template for Covid-19 was also developed [8], which includes archetypes for assessments, clinical background, treatment and discharge, and is available in the CKM.[11]

In openEHR, archetypes typically model healthcare concepts, such as blood pressure, while templates typically model forms, documents and messages. Archetypes are defined in ADL (Archetype Description Language) and can be queried using AQL (Archetype Query Language), analogous to how SPARQL can be used to query RDF (Resource Description Framework) triple stores. An essential difference between using openEHR and RDF to represent clinical knowledge, is the kinds of system interoperability they aim at. openEHR intends to enable interoperability in health care systems with a focus on electronic health care records, whereas RDF is domain independent and intended to be used at web scale.

The Semantic Web can be thought of as a suite of semantic web technologies together with Linked Data, a set of best practices for sharing data on the web. These semantic web technologies allow the creation of data stores on the web, the building of vocabularies, and the writing rules for handling data. Linked Data are supported by technologies such as RDF, SPARQL, OWL, and SKOS.[12] In RDF, a description of a resource[13] is represented as a number of triples, each of which codifies a statement about semantic data and consists of a *subject*, *predicate* and *object* [4, 1]. These subjects, predicates and often also the objects[14] themselves are URIs of concepts that reside in precise formal vocabularies and ontologies. RDF, therefore, relies on *semantics by reference*.[15] As the abstract data model of the Semantic Web, RDF is considered one of the dominant graph technologies currently driving semantic computing over web-scale distributed data.

More specifically, semantic web technologies and Linked Data, combined with big data analytics, have become key to making patient data semantically interoperable and to helping create predictive models on how epidemics might spread

[9] https://specifications.openehr.org/releases/BASE/latest/architecture_overview.html
[10] https://ckm.openehr.org/ckm/archetypes/1013.1.4399
[11] https://ckm.openehr.org/ckm/templates/1013.26.291
[12] https://www.w3.org/standards/semanticweb/
[13] A web resource, or simply resource, is any identifiable thing, whether digital, physical, or abstract. Resources are identified using Uniform Resource Identifiers (URIs)
[14] The object can also be a literal. Literals are used for values such as strings, numbers, and dates. See https://www.w3.org/TR/rdf11-concepts/
[15] https://www.w3.org/TR/rdf-mt/

around the world[16] [13]. It is therefore important to ensure that patient data, also for Covid-19, is exposed as Linked Data as accurately and as seamlessly as possible.

Grammatical Framework (GF) [16] is a programming language for grammar engineering, which uses a semantic interlingua for multilingual machine translation. It has become the *de facto* standard in multilingual controlled natural language applications [17]. GF has been used in a number of knowledge representation projects. For example GFMed [11] is a question answering system for biomedical interlinked data. It employs GF grammars for a controlled language targeted towards biomedial information and the SPARQL query language. In [3] an approach to multilingual ontology verbalisation of controlled language based on GF and the *lemon model*[17] is presented.

A number of knowledge representation projects in the indigenous South African languages have been reported. For example, [6] discuss isiZulu verbalisation patterns for basic logic constructs and devised algorithms to generate grammatically correct isiZulu sentences.

To the best of our knowledge, AwezaMed Covid-19 is unique in that it is the only speech-enabled mobile application currently in existence that supports multilingual, multimodal machine translation to all South African languages based on a semantic interlingua. This paper describes how knowledge representation can be added.

3 The speech-to-speech translation architecture

In this section, we discuss the choice of a semantic interlingua translation architecture for our specific use case, namely facilitating screening of patients for Covid-19, and the effect this has on the way the application can facilitate multilingual communication between HCPs and patients.

3.1 Choosing a suitable translation architecture

Deep learning techniques have made enormous progress in the last few years and are the state-of-the-art in machine translation for the large languages of the world. They rely on the very large amounts of data that have become available for many languages. However, all the official South African languages excluding English are so-called under-resourced languages, which means that the same amounts of data are not available – and in many cases do not exist – as for the larger languages of the world. This means that other techniques are worth investigating if a specific use case could benefit from it.

Available MT capability for English to other South African languages The health care domain requires a high level of accuracy in machine translation. Achieving appropriate accuracy using state-of-the-art data-driven machine

[16] https://www.ontotext.com/blog/linked-data-solutions-in-healthcare/
[17] https://lemon-model.net/

translation techniques depends on two major factors: the availability of domain appropriate parallel data for each language pair, and the linguistic similarity between the source and target languages. [12] report BLEU scores for translation from English to isiXhosa and isiZulu of 37.11 and 44.07, respectively. However, these scores are the result of training and testing on the single domain JW300 corpus. When the same systems were tested on the Autshumato test corpus of government documents [14], BLEU scores dropped dramatically to 1.42 and 1.56, respectively.

In contrast to approaches that rely on large amounts of data, a grammar-based approach can provide a more controlled form of domain appropriate translation, where coverage is constrained, but a high level of accuracy is guaranteed.

The goal of GF is reduce the amount of effort and time traditionally required by rule-based machine translation [16], so that building multilingual domain specific machine translation systems for real world applications is feasible.

Speech-to-speech and mobile application integration GF is well-suited to supporting multimodal, multilingual translation applications [2]. Its diverse module system makes it possible to linearise the same utterance in different languages, as well as different formats for the same language that enables support for the formatting conventions of ASR and TTS.

Given the way a grammar limits the domain of supported utterances, it must also support a touch input modality that can present the translatable content of the application to the user in a compact and intuitive way. This acts both as a mechanism for familiarising users with the domain covered by the application, as well as a fallback mechanism for when speech recognition fails.

Different roles of participants in a screening The nature of the utterances used by HCPs during a screening is relatively structured and predictable. This is especially true when the domain is limited to a subdomain of health care, such as screening for Covid-19. The grammars in AwezaMed were developed in close consultation with various HCPs to cover relevant and useful content.

In contrast to HCPs, who have specific domain knowledge and are responsible for driving the communication during a screening in order to arrive at a correct finding, patients are not domain experts. Their answers to questions arise from their experiential knowledge of their health, and may range widely in terms of detail, focus and applicability. Hence, constrained domain grammars are not suitable to model typical patient utterances.

3.2 A communication model for grammar-based machine translation

Limiting speech-to-speech translation to only cover utterances uttered by the HCP has an obvious impact on the kinds of utterances that will enable HCPs to conduct an entire screening using the application. Specifically, HCPs will have to understand the patient responses without the help of the application.

Hence, the communication model of the AwezaMed application limits almost all utterances to binary questions (i.e. requiring only a "yes" or "no" answer from the patient). In reality, therefore, the content of the application is such that statements of fact are presented to the patient in the form of questions which they can confirm or deny. The HCP need only understand the words for "yes" and "no" in the patient's language, or understand gestures such as nodding or shaking of the head, to establish relevant observations about the patient's health.

We see, therefore, how an analysis of the specifics of the use case leads to the choice of architecture and communication model, namely a GF-based domain grammar that translates binary questions posed by the HCP in the source language to a target language that the patient understands.

4 Semantic interlingua machine translation for Covid-19 screening

We turn now to the mechanism employed by GF to support domain grammars, namely a semantic interlingua architecture. The goal of a GF domain grammar, known as an application grammar, is to start with the semantics of the domain, and express it in one or more natural languages [15]. The semantics is defined via categories and functions in an abstract syntax, while one or more concrete syntaxes define how such categories and functions are linearised as strings.

Developing a concrete syntax therefore involves defining a *linearisation category* in the form of a record for each category in the abstract syntax, and a *linearisation function* for each function in the abstract syntax, which defines how types of records are combined into new ones. Parsing is based on inversion of the linearisation rules in a non-trivial way [15]. GF can therefore be seen as a multi-source, multi-target compiler [9], where any string in one of the concrete syntaxes of the grammar can be translated to any other language by parsing the source utterance string into an abstract syntax tree and linearising the tree into a string in the target language.

The way in which semantically equivalent translation is achieved can be understood by considering Wittgenstein's notion of a language game [15]. An application grammar is effectively the definition of a specific language game, where translation is possible if the same language game can be played in both the source and target languages. Stated differently, starting with the semantics of a domain, translation is possible if the same abstract syntax tree, capturing some meaning in a specific domain, can be expressed as natural language utterances in two or more languages. Because translation is achieved via a semantic interlingua in the form of the abstract syntax, as long as the utterance in the source language represents the intended meaning of the user in the context of the domain, the user can be confident that the translation of the utterance represents the same meaning in the target language.

4.1 Implementing multilingual, multimodal machine translation for a mobile application

The application contains four application grammars, namely *Symptoms*, for asking about symptoms associated with Covid-19, *Medical History*, for establishing the presence of possible comorbid conditions, *General History*, covering allergies and substance usage, and *Covid-19*, which is mainly for relaying information and instructions related to Covid-19. For the purpose of showing how the grammar enables translation via a semantic interlingua, our focus will be on the *Symptoms* grammar.

Semantic trees in the abstract syntax The *Symptoms* application grammar allows the HCP to ask questions about Covid-19 related symptoms, including whether a patient has a certain symptom, whether the symptom started more, less or about a certain number of days ago, whether the symptom is persistent and whether the symptom is worsening. Fig. 1 shows an example of an abstract syntax tree, which is expressed in English as "Did the fatigue start about two days ago?". In each node label, the function name used to construct that particular constituent appears to the left of the colon, while the category type of the constituent appears to the right.

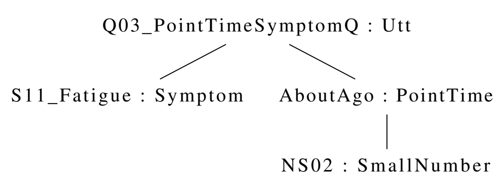

Fig. 1: Semantic tree for "Did the fatigue start about two days ago?"

Multiple languages and modes Each concrete syntax provides rules for linearising semantic trees to strings. A detailed discussion of how GF provides parameters and tables to implement grammar rules is beyond the scope of this paper, and the interested reader is referred to [16]. It suffices to say that the implementation of a concrete syntax for a specific language enables the GF runtime to generate strings in that language in a compositional way to represent the meaning of an abstract syntax tree.

The difference between the way the tree in Fig. 1 is expressed in English and isiZulu can be seen by comparing Fig. 2 and Fig. 3, which show how the semantic constituents of the abstract syntax tree are linearised into strings in the two languages. In order to facilitate the discussion in Section 5, the strings are colour coded, with strings in violet contributed by the `Symptom` category, green by the `PointTime` category, orange by the `SmallNumber` and black by the `Utt` category, which is the start category.

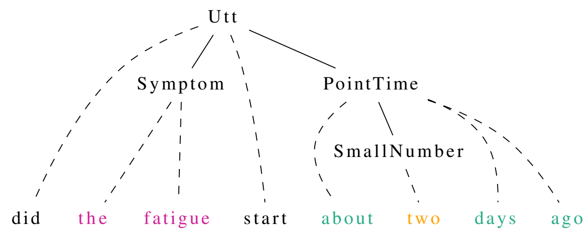

Fig. 2: Parse tree for expressing "Did the fatigue start about two days ago?" with ASR formatting conventions

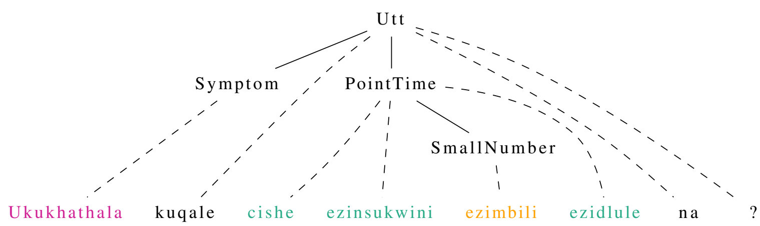

Fig. 3: Parse tree for expressing "Did the fatigue start about two days ago?" in isiZulu

Grammar-driven dynamic screen Note that the English string in Fig. 2 contains no punctuation or capitalisation. This is because the grammar contains a concrete syntax specifically for defining the appropriate English strings with formatting that follows the conventions of the ASR component of the application. For the isiZulu strings, which must be displayed on the screen and also serve as input to the text-to-speech component, capitalisation and punctuation must be included.

However, in addition to the speech modalities supported by the grammar, the touch modality must also be supported. To this end, another version of the English concrete syntax exists that adds markup to a capitalised and punctuated English string, which can be used to determine which parts of the string must be "live", in the sense that the user could click on it to change it. Fig. 2 shows the parse tree for the example utterance generated by this concrete syntax (slightly simplified for readability), while Fig. 5 shows how the user interface uses the information encoded in the marked up string. Each function in the grammar that produces an `Utt`, which is the start category of the grammar, corresponds to a so-called *dynamic utterance*, which could be thought of as a dynamic, grammar driven template for presenting many utterances on a single screen in an intuitive way.

5 Knowledge representation of patient responses

How does this speech-to-speech architecture, chosen due to the constraints provided by the use case, enable knowledge representation of patient responses?

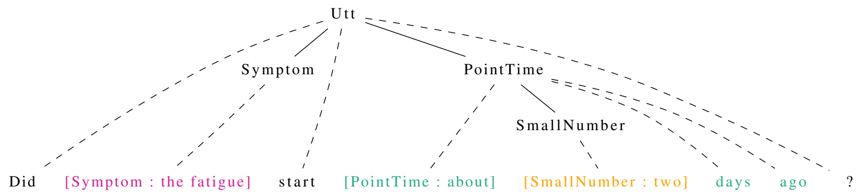

Fig. 4: Simplified parse tree for expressing "Did the fatigue start about two days ago?" with dynamic utterance markup

(a) Dynamic utterance (b) Making a selection

Fig. 5: Screen elements derived from dynamic utterance markup

5.1 From questions to observations

When a binary question is put to the patient, and the patient answers in the affirmative, the semantic tree of the utterance effectively represents an observation about the patient. For example, if the question "Did the fatigue start about two days ago?" is confirmed by the patient, the statement "The patient reports fatigue, which started two days ago" could be noted as an observation.

Practically speaking, the HCP might use the application to input some binary question in English which the app would then translate to the appropriate target language. When the patient answers in the affirmative, the HCP would want to capture this information in an electronic health record. In order to do this, the application should be extended with two specific features:

– Implementation of a patient information section for creating and updating patient entities in a suitable data repository. This would allow each screening session to add the observations obtained to a specific patient's health record.
– Implementation of a check box next to each dynamic utterance in the application, which would allow HCPs to immediately mark some utterance as being confirmed by the patient.

The first features concerns the management of patient EHRs in healthcare systems. Our focus is on to the second feature, which would allow capturing of clinical observations in a seamless and semantically interoperable way.

5.2 Generating knowledge representations with GF

In openEHR, an OBSERVATION is a type of ENTRY suitable for symptoms, test results and other similar clinical information. In this section we show how the *Symptoms* grammar can be extended to generate knowledge representations in the form of openEHR OBSERVATIONs, as well as RDF triples that similarly represent the observation.

Since our goal is to describe the mechanism by which a semantic interlingua can be used to express the confirmation of natural language binary questions as formal knowledge, we focus on generating just those snippets of code in both formalisms. Representing administrative information, that would connect these observations to patients is outside the scope of this paper.

Representing knowledge according to openEHR specifications can be done in JSON, and similarly, RDF triples can be encoded using Turtle. In order to generate JSON and Turtle code that represent observations, we added two concrete syntaxes to the existing translation architecture. Therefore, in addition to multiple languages and speech modalities, the grammar was extended to support two formal knowledge representation formalisms.

In the case of openEHR, the JSON code generated by the grammar would be integrated into a COMPOSITION structure that includes all necessary administrative information, while the RDF triples generated by the grammar can be added to a triple store that similarly connects the observation to the relevant administrative information.

As with any other concrete syntax, linearisation categories and linearisation functions were be defined in order to generated JSON and Turtle code. Given the context-free nature of JSON and Turtle, records containing a single string sufficed as linearisation categories, and straight-forward token concatenation was employed in the linearisation functions. Fig. 6 and Fig. 7 show strings generated by the two new concrete syntaxes, with the same colour coding as before.

The openEHR JSON concrete syntax generates code consistent with the Covid-19 symptom archetype, and uses the SNOMED-CT vocabulary to refer to specific symptoms. The RDF Turtle concrete syntax uses SNOMED-CT in the same way, in addition to the FHIR ontology.

In both code snippets, the string `<_SYSDATE_ - 0000-00-02>` is generated by the grammar with the intent that the host application resolve this based on the system time of the mobile device. The host application must also generate a unique identification number for each observation. Due to the differences in how observations are included in the different data repositories, the Turtle snippet includes the string `_OBSERVATION_ID_` that must be replaced. In the case of the openEHR JSON code, this string occurs in a different part of the `COMPOSITION` data structure, and is therefore not shown here.

The reason that the `onset_time` concept in the Turtle code is defined separately as shown, is because RDF provides more freedom to define time related concepts, via the W3C Time ontology,[18] than is possible given the Covid-19

[18] https://www.w3.org/TR/owl-time/

```
{ "content":
  { "archetype_node_id": "openEHR-EHR-CLUSTER.symptom_sign-cvid.v0",
    "type": "OBSERVATION",
    "name": { "value": "Covid-19 symptom" },
    "archetype_details":
    { "archetype_id": { "value": "openEHR-EHR-CLUSTER.symptom_sign-cvid.v0" } },
    "data":
    { "archetype_node_id": "at0001",
      "type": "ITEM_TREE",
      "name": { "value": "components" },
      "items": [
      { "archetype_node_id": "at0001.1",
        "type": "ELEMENT",
        "name": { "value": "Symptom/Sign name" },
        "value":
        { "type": "DV_CODED_TEXT",
          "value": "Fatigue",
          "defining_code":
          { "terminology_id": { "value": "SNOMED-CT" },
            "code_string": "84229001" } } },
    { "archetype_node_id": "at0152",
      "type": "ELEMENT",
      "name": { "value": "Episode onset" },
      "value":
      { "type": "DV_DATE_TIME",
        "value": "<_SYSDATE_ - 0000-00-02>" } } ] } } }
```


Fig. 6: Generated JSON snippet of the tree in Fig. 1

```
@base <http://example.org/> .
@prefix rel: <http://www.perceive.net/schemas/relationship/> .
@prefix fhir: <http://hl7.org/fhir/> .
@prefix time: <http://www.w3.org/2006/time#> .
@prefix sct: <http://snomed.info/id/> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

<_OBSERVATION_ID_>
    a [ fhir:ClinicalImpression sct:84229001 ] ;
    sct:405795006 <onset_time> .
<onset_time>
    a time:Instant .
    time:hasTime "<_SYSDATE_ - 0000-00-02>"^^xsd:dateTime .
```

Fig. 7: Generated Turtle snippet of the tree in Fig. 1

symptom archetype available in openEHR. In the latter, the symptom onset can be given as a time instant of type `DV_DATETIME`. However, the Time ontology allows more complex time concepts, with properties `time:before` and `time:after`. These properties can be used to represent binary questions in the grammar such as "Did the fatigue start more than two days ago?" and "Did the fatigue start less than two days ago?". The latter example is represented as follows, and is generated when the `PointTime` constituent is created using the `LessThanAgo` function instead of the `AboutAgo` function:

```
<onset_time>
    a time:Instant .
    time:after "<_SYSDATE_ - 0000-00-02>"^^xsd:dateTime .
```

Depending on the choice of knowledge representation formalism, the application could disable check boxes for any instances of dynamic utterances it cannot represent. Another solution might be for a clinician to contribute an openEHR archetype for symptoms that contains additional item elements for expressing relative time concepts.

6 Discussion

Electronically capturing our observation, namely "The patient reports fatigue, which started two days ago", would typically require that an HCP select the symptom, fatigue, from a list on a screen, and select the date of two days ago from a calendar widget. This is time consuming and interferes significantly with the interaction between the HCP and patient.

The essence of our contribution is in providing a proof-of-concept implementation of core components that would enable a seamless process for capturing patient information, even in cases where the HCP and patient do not share a common language. Relatively complex information can be captured by speaking it as a question, letting the application translate the input to speech in a different language, and checking a box when it is confirmed. In this way, we let the communication act facilitated by the mobile application do the heavy lifting of establishing the semantic content of the information to be captured.

Our implementation covers 396 unique utterances. We have implemented four utterance types, which reference 12 Covid-19 related symptoms, three different ways of expressing relative time, and 10 small numbers for counting days. In implementing the knowledge representation concrete syntaxes to support these utterances, concepts and formal information models that correspond to the utterances were identified within the openEHR and HL7 frameworks. This includes the openEHR archetype for Covid-19 symptoms, the SNOMED-CT vocabulary and several other ontologies.

In order to transform the existing AwezaMed app into a knowledge representation aid for Covid-19 screening as described in this paper, the steps identified in [13] could be implemented in two phases:

- **Ontology Development** Other utterances supported by the grammar must be analysed and the concepts they express must be associated where possible with existing knowledge representation concepts. In cases where the necessary terms, classes, properties and constraints do not exist, ontology development is required.
- **Semantic Data Creation** This phase entails the extension of the application with the two features mentioned in Section 5.1, as well as its integration with real data repositories containing semantic data of real patients.

7 Conclusion

The starting point of our work was an existing speech-to-speech translation application, which was implemented using a semantic interlingua, chosen due to the constraints of the use case. We extended this architecture to provide a way to formally represent the knowledge gained while using the application.

This extension was implemented as additional concrete syntaxes in the application's GF translation system: in addition to linearising (or parsing) a semantic tree as a binary question in multiple languages, it was also linearised by the grammar into two knowledge representation formalisms.[19]

By implementing the extension according to established formalisms, namely openEHR and RDF, which exist within larger frameworks for representing knowledge in health care systems and on the web, we have shown how an application such as AwezaMed could integrate with such systems to contribute to the acquisition, publication and interoperability of health care information. This, in turn, would serve to enable better understanding and improved responses to viral pandemics such as Covid-19.

[19] The natural language formulation of the observation, namely "The patient reports fatigue, which started about two days ago" could also be linearised by various concrete syntaxes, resulting in multilingual human readable statements about the observations made during the screening.

References

1. W3C RDF 1.1 Primer (2014)
2. Angelov, K., Bringert, B., Ranta, A.: Speech-enabled hybrid multilingual translation for mobile devices. In: Proceedings of the Demonstrations at the 14th Conference of the European Chapter of the Association for Computational Linguistics. pp. 41–44 (2014)
3. Davis, B., Enache, R., Van Grondelle, J., Pretorius, L.: Multilingual verbalisation of modular ontologies using gf and lemon. vol. 7427 (08 2012)
4. Heath, T., Bizer, C.: Linked Data: Evolving the Web into a Global Data Space. Morgan & Claypool (2011), iSBN: 9781608454303
5. Kalra, D., Beale, T., Heard, S.: The openehr foundation. Studies in health technology and informatics **115**, 153–173 (2005)
6. Keet, M., Khumalo, L.: Toward a knowledge-to-text controlled natural language of isizulu. Language Resources and Evaluation **51**(1), 131–157 (Mar 2017)
7. Leslie, H., Heard, S., Garde, S., McNicoll, I.: Engaging clinicians in clinical content: herding cats or piece of cake? In: MIE. pp. 125–129. Citeseer (2009)
8. Li, M., Leslie, H., Qi, B., Nan, S., Feng, H., Cai, H., Lu, X., Duan, H.: Development of an openehr template for covid-19 based on clinical guidelines. J Med Internet Res **22**(6), e20239 (Jun 2020). https://doi.org/10.2196/20239, http://www.jmir.org/2020/6/e20239/
9. Listenmaa, I.: Formal Methods for Testing Grammars. Ph.D. thesis, Gothenburg, Sweden (2019)
10. Marais, L., Louw, J.A., Badenhorst, J., Calteaux, K., Wilken, I., van Niekerk, N., Stein, G.: Awezamed: A multilingual, multimodal speech-to-speech translation application for maternal health care. In: 2020 IEEE 23rd International Conference on Information Fusion (FUSION). pp. 1–8. IEEE (2020)
11. Marginean, A.: Question answering over biomedical linked data with grammatical framework. In: Cappellato, L., Ferro, N., Halvey, M., Kraaij, W. (eds.) Working Notes for CLEF 2014 Conference. Sheffield, United Kingdom (Sep 2014), https://hal.inria.fr/hal-01086472
12. Martinus, L., Webster, J., Moonsamy, J., Jnr, M.S., Moosa, R., Fairon, R.: Neural machine translation for South Africa's official languages (2020)
13. McGlinn, K., Hussey, P.: An Analysis of Demographic Data in Irish Healthcare Domain to Support Semantic Uplift. In: ICCS 2020, LNCS 12140 (2020)
14. McKellar, C.A., Puttkammer, M.J.: Dataset for comparable evaluation of machine translation between 11 South African languages. Data in Brief **29**, 105146 (2020). https://doi.org/https://doi.org/10.1016/j.dib.2020.105146, http://www.sciencedirect.com/science/article/pii/S2352340920300408
15. Ranta, A.: Type theory and universal grammar. Philosophia Scientiæ. Travaux d'histoire et de philosophie des sciences (CS 6), 115–131 (2006)
16. Ranta, A.: Grammatical Framework: Programming with Multilingual Grammars. CSLI Publications, Stanford (2011), iSBN-10: 1-57586-626-9 (Paper), 1-57586-627-7 (Cloth)
17. Safwat, H., Davis, B.: A brief state of the art of CNLs for ontology authoring. In: Davis, B., Kaljurand, K., Kuhn, T. (eds.) Controlled Natural Language. pp. 190–200. Springer International Publishing, Cham (2014), iSBN: 9783319102238