|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ITU Logo | INTERNATIONAL TELECOMMUNICATION UNION  **TELECOMMUNICATION STANDARDIZATION SECTOR**  STUDY PERIOD 2017-2020 | | FG-AI4H-A-020 | |
| **ITU-T Focus Group on AI for Health** | |
| **Original: English** | |
| **WG(s):** | | N/A | Geneva, 26-27 September 2018 | |
| **DOCUMENT** | | | | |
| **Source:** | | Ada Health GmbH | | |
| **Title:** | | Towards a potential AI4H use case "diagnostic self-assessment apps" | | |
| **Purpose:** | | Discussion | | |
| **Contact:** | | Henry Hoffmann Ada Health GmbH Germany | | Tel: +49 177 6612889 Email: [henry.hoffmann@ada.com](mailto:henry.hoffmann@ada.com) |
| **Contact:** | | Andreas Kühn Ada Health GmbH Germany | | Tel: +49 30 60031987 Email: [andreas.kuehn@ada.com](mailto:andreas.kuehn@ada.com) |

|  |  |
| --- | --- |
| **Abstract:** | This document explores the potential AI4H standardization use case "diagnostic self-assessment apps". It points out the need for a standardization to then give an overview of the relevant aspects that could be considered for setting up a standardization process for this use case. |

# Rationale and Scope

The WHO estimates the shortage of Global Health workers to increase from 7.2 million in 2013 to 12.9 million by 2035. This shortage is driven by not only population increases in low and middle income countries, but also due to the increased health-related needs in high income countries. At the same time, patient-centered diagnostic decision support systems (DDSSs), also called "self-assessment-apps" or "symptom checkers", have become widely available. The main focus of these apps is to give users tangible health advice for their health related problems and to provide guidance on relevant next steps. These next steps include self-care options, pharmacies, and referral to health facilities. While patient empowerment by such apps has great potential to improve the outcome for patients by providing earlier diagnosis of conditions and more efficient care-navigation though the health system, the lack of any standardization makes it difficult for organizations like the WHO, governments, and other key players to adopt self-assessment-apps as part of their solutions for addressing global health challenges. The purpose of this document is to explore the most relevant aspects of diagnostic self-assessment apps to establish a theoretical standardization framework.

# Diagnostic Self-Assessment-Apps Structure Overview

The main goal of diagnostic self-assessment-apps (DSAAs) is to provide explanations and guidance for health problems that a user is currently experiencing. The overall structure of an assessment in this case can be defined as:

1. **User profile creation:** The collection of age, sex and medical background information.
2. **Presenting complaints**: The acquisition of the presenting complaint(s) the user wants advice for.
3. **Evidence gathering**: The collection of additional evidence.
4. **Result presentation**: The presentation of the assessment result.

While profile creation and user onboarding is fairly straightforward and similar amongst DSAAs, the remaining steps vary considerably as apps take different technological approaches to diagnostic decision support. **Presenting complaints** (PCs) are the key complaints the user seeks help for. At the beginning of each assessment, apps collect PCs to direct the symptom assessment towards a relevant explanation and to contextualize the assessment report. Examples for PC collection techniques are:

* Browsing lists or trees of symptoms
* Symptom search solutions
* Free-text understanding

It is noteworthy that not all systems allow multiple presenting complaints. This needs to be considered if testing multiple systems. Since the collection of PCs is crucial for the quality of the overall assessment, the mapping of search queries and free text to correct symptoms should be standardized.

After entering PCs, a DSAA usually **collects additional evidence** to help confirm suspected diagnoses, to take possible differentials into account, and to rule out others. Approaches to collect additional health-related evidence include:

* Static, hierarchical questionnaires or symptom search
* A dialog / question flow actively asking for symptoms
* A chatbot understanding free-text and actively asking for symptoms

An important difference between available systems is whether users can provide negative evidence and whether they can skip questions. In most systems the total number of questions is limited. It is important to understand that AIs that engage in a dialog with the user are more difficult to test since they can ask potentially for even seemingly irrelevant symptoms, while a given test case cannot contain all the answers to all possible symptoms.

The starkest differences between self-assessment apps are the types of advice they provide as part of the **results presentation**. The most important assessment outputs include:

* **Pre-Clinical-Triage:** Classification of the severity of the users health state and what to do next (e.g. to seek immediate emergency care or that self-care, possibly with pharmacy support is sufficient). This might also consider other conditions not related to the PCs.
* **Pre-Diagnosis:** Listing of one or more possible causes of the PCs. These causes may be accompanied by a quantification measure, such as a likelihood or score.
* **Other-Diagnosis:** Listing one or more other conditions not related to the presenting complaints, but possible from the answered questions. These may also be accompanied by a quantification measure. It is possible that a user seeks help for a trivial problem while the dialog identifies a more serious constellation. In this case DSAA should give the correct advice for the PC while still flagging other potential, hazardous differentials.
* **Explanations:** An explanation of how factors (age, sex, etc.), PCs, and symptoms are related to the suggested conditions, possibly including a quantification of the contribution.
* **Next step advice:** A result-dependent suggestion of reasonable next steps. This can include therapy options, referral to health facilities or to other health providers

DSAAs usually provide a mix of the described outputs. Some systems include also the possibility of a healthy user experiencing an isolated symptom without an underlying causing disease (e.g. idiopathic nose bleed).

# Assessing the Diagnostic Quality of DSAAs

There are several ways to test the diagnostic quality of DDSSs. Besides rule-based systems implementing protocols such as ICCM/IMCI, DSAAs based on stronger AIs are mostly tested using labeled test-cases with the expected output and true underlying cause of disease.

## Test Case Structure

To benchmark DSAAs, test-cases require the following components that reflect the different phases of an assessment and possible outputs outlined in section 2:

* **User profile data:** Information the DSAA usually ask for during user-onboarding/profile creation. This usually includes basic demographics such as age, biological sex, and medical history.
* **Presenting complaints:** Health problems that the user seeks advice for.
* **Additional symptom evidence:** All other symptoms not reported as PCs but collected afterwards. This includes negative evidence.
* **Expected pre-clinical-triage:** Expected triage output of a DSAA. This could also possibly be a range of triage results or a list of "excludes" or "includes".
* **Expected PC causes:** Conditions suspected or considered valid causes of the presenting complaints. They might be annotated with scores or classes (e.g. "primary", "acceptable", "essential differential") that are then considered within the quality measures.
* **Expected other conditions:** Non-PC related expected conditions based on all collected health information

Since demand for AIs to explain their results is growing, it is also desirable to have a formalized way to test explanations. However, given the diversity of AI approaches, it appears infeasible to standardize this part of the test cases. More reasonable is a mechanism to assess the plausibility or consistency of an explanation as part of the benchmarking system during one of the later iterations of an ITU standardization for DSAAs. Additionally, an expectation for valid "Next Step Advice" might not be part of the cases, since this is specific to the individual provider and the context of the user. The list above may need to be extended to any DSAA output that requires standardization.

## Quality Measures

For assessing the quality of a DSAA, all reference cases need to be manually entered. This includes engaging in the automated dialog between the testing system and the DSAA. Both the dialog and the final output are recorded and then assessed to achieve quality measures. The measures usually map each case to an individual result and also calculate performance totals on a per-condition and overall basis. The measure definitions are crucial to standardization because they define what a "good" DSAA is. It will likely be necessary to define a whole panel of measures for benchmarking different aspects of DSAAs and their applications in different scenarios. Necessary considerations to design the panel include but are not limited to:

* **Prior scaling:** All conditions are more or less equally represented in the test sets. This gives rare diseases a much higher weight in the aggregation of the total scores. While this is desirable to make sure that all disease models perform well, in some cases it is more important to measure the net performance of a DSAA in real world scenarios. In this case the aggregation function needs to scale the individual cases results with it's expected top match prior probability in order to get the mathematically correct expectation-value for the score. For example, errors on common-cold cases need to be punished harder than errors on cases of rare diseases that only few people on the planet suffer from.
* **The rank position:** In case the expected top-match is not in the first position, the actual position might be part of the scoring. This could include the probability integral of all higher ranking conditions or the difference between the top scores and the score of the expected disease.
* **Medical distance of the top matching diseases to the expected ones:** In case the expected top match is not the first position and the listed conditions are not in the "expected other conditions", the medical distance between the expected conditions and actual conditions could be included in the measure. This distance could utilize available hierarchy information from the condition ontology.
* **The role of the secondary matches:** Since DSAAs usually present multiple possible conditions, even if the top match is correct the qualities of the other matches need to be considered as well. For example, the highly relevant differentials that should be ruled out are much better secondary diagnoses than random diseases.
* **Number of questions:** For dialog-based systems, the number of questions asked might also be relevant for the scoring. It may be necessary to split the DSAA into different subclasses by the number of questions they ask.
* **"Expected PC causes" label:** The labels ("acceptable differential", "essential differential", etc. ) of the conditions mentioned in the "expected PC causes" fields of the test case should be included within the measures. For instance, if a condition marked as an "essential differential" for a case is contained in the result set, the case should score better.
* **Performance:** Even if not part of the diagnostic quality, there should be measures considering the average time for calculating the next questions and the time for the calculation of the final result. This may also help to make sure the tested systems are indeed DSAAs rather than manual approaches that are not within the scope of the current standardization.

It is worth noting that when possible, expectations should be preferably expressed as part of a case, rather than as part of the measures. For instance, if dangerous non-PC-related diseases should be considered, they should be mentioned in the "expected other conditions" rather than being weighted into some measure. This is also true for special, important presenting complaints reflected in the dedicated "expected PC causes" part of the test-cases. The definition of quality measures is quite complex, but crucial, to optimize DSAAs for their use cases and to assess a provider's technology.

# A Framework Benchmarking DSAAs

Most systems with a stronger AI component run on server clusters hosted by one of the major cloud computing providers. In contrast to e.g. testing a codec they can therefore not easily be handed over for independent testing. Another important aspect is that for most companies, their AI and medical knowledge makes up a significant portion of their core Intellectual Property (IP) and cannot be given to third parties. The rapid development of technology (e.g. new releases of the AI and underlying knowledge in regular cycles) and the diversity of AI approaches makes them effectively "black-boxes" that need to be benchmarked by the output response to given inputs. The main focus of standardization should be to provide an external technological and methodological framework for assessing the quality of the available AIs in an objective and transparent way.

One approach for testing AIs under the given constraints could be to implement a dedicated API endpoint. There could be central testing service hosted by the ITU where participants can register their AI, define their feature set to benchmark, then possibly give some background information and the URL of the API endpoint. The testing service could provide a web interface to run tests against the providers API endpoints at any time, showing detailed total statistics as well as an analysis for each case. Since high-quality, labeled case data is an expensive and rare resource, individual testing should run against all the data already used in official benchmarking sessions. It is the responsibility of the provider to not overfit to test data. In theory, the AI could learn to recognise the test cases and perform with 100% quality, while having 0% quality on "real world" cases.

For **official benchmarking**, a new set of test cases unknown to all AI providers must be created. The data for official benchmarking might contain a small fraction that is available to the providers before the official testing, especially if it contains new features. On a regular basis, all registered AIs are then tested based on this new data set. This benchmarking might be split in three phases. In the first phase, the testing server collects the outputs from all the AIs. During the second phase, the cases are made public and providers can analyse the results or make comments and suggestions (e.g. an important essential differential is missing in a test case). This phase might lead to discarding cases that are proven medically wrong. In the final phase, official measures are then calculated based not on the possibly adjusted case labels, but still using the outputs collected in the first phase. The benchmarking results can then be published in the publicly available section of the ITU DSAA testing service. It is necessary to discuss whether providers should be allowed to not publish their results or to limit results to totals rather than individual cases.

1. **Individual testing:** Providers may test their DSAA against already known cases at any time.
2. **Test set creation:** A secret benchmarking dataset is collected and labeled with the expected outputs
3. **Phase 1:** The output of all DSAAs for benchmarking test cases is collected
4. **Phase 2:** Providers are given the chance to analyse the results and to give feedback on questionable case annotations
5. **Phase 3:** The official benchmarks are calculated based not on the adjusted case labels and the DSAA outputs collected in phase 1.
6. **Publication:** The benchmarking results are officially published. The providers might have the right to opt out.

While AIs develop quite fast, the creation of new test data takes time. At the same time, larger integration projects of AIs into health systems will probably start only on a quarterly basis, so having official testing every 3 or 6 months might be a good starting point.

Since recent experience in the car industry has highlighted the ability to manipulate tests, the whole benchmarking framework needs to be designed as safely and resiliently as possible. This requires careful planning, and some necessary points to consider are:

* **Case by case testing:** The official benchmarking should be processed case by case all and at the same time. This way, having multiple DSAA registered does not give a provider an advantage by knowing the test-set from one of its other apps.
* **Assume storage of all test cases:** For every test case ever sent to a DSAA endpoint, it must be assumed that it was stored and used for improving the AI, and hence can never be used again for any official benchmarking.
* **Timeouts:** There should be timeouts that make sure that the process cannot be faked using real doctors.
* **Inter operator correlation:** There should be statistical tests in place making sure that one provider is not secretly using the engine of an other provider.
* **API endpoint authentication:** DSAA endpoints should use https, oAuth authentication and only accept requests by the official testing service.
* **Two factor authentication:** The web interface of the testing service should use two-factor authentication.

# General Test-Set Requirements

The most crucial part of DSAA benchmarking is the regular creation of a test-set. The general requirements for such a test-set might include:

* **Covering all conditions:** There should be at least 1-2 two typical cases for every condition. This gives transparency on which conditions are currently supported by which provider, shows improvements for individual conditions that might have underperformed in the last benchmarking, and makes sure that the results are representative.
* **Atypical presentations :** The test-set should also include atypical presentations of conditions, as they are more difficult to diagnose in the clinical setting, have a high risk of being misdiagnosed, and therefore a higher potential benefit to improve health care.
* **Emergency and Red Flag Cases**: In the (pre-)clinical setting, some presentations, such as acute radiating chest pain in older smokers, may require immediate emergency care and an DSAA has to recognize almost all of these emergency cases and give the right advice. In other scenarios, life threatening conditions may underlie certain symptoms (so called "red flag symptoms") and it is essential that this situation is recognized by the DSAAs.
* **Covering all stages:** Some diseases change considerably over time. Usually, diagnosis in the terminal stage is too late for both doctors and AIs (e.g. late-stage Parkinson's disease). It is imperative to catch early signs and symptoms of disease.
* **Strong focus on "moment of first diagnosis":** While the press and potential users tend to test DSAAs with typical late-stage cases, the main use case of DSAAs is to identify a condition during the initial presentations of relevant symptoms, before a condition has already advanced into later stages. Therefore, the majority of test-cases should focus on early/initial stages of diseases.
* **Strong focus on patient reportable symptoms:** In contrast to clinical DDSSs, the scope of DSAA use cases relies on patient-facing systems. The cases should first and foremost contain symptoms that laymen can easily assess and understand by themselves.
* **Not known to the providers:** To have a fair benchmarking process, it is clear that the providers must not know the test set in advance. The data can therefore not be taken from a public source nor for a repository one of the providers has access to. Given that the purpose of standardization is to allow better integration into other solutions (including even clinical contexts), it will become more and more difficult over time to collect unseen datasets.

# Ontologies for DSAA Input and Output

Another essential requirement for reference cases is that they describe inputs and outputs using agreed-on ontologies. A symptom ontology is needed to properly describe inputs. Its quality is essential to the quality a DSAA can deliver. The criteria for selecting the right symptom ontology may include:

* **Symptom, factors and findings:** The symptom ontology should cover risk factors and medically relevant background details like age, sex, medical history, family history, genetic predisposition, living conditions, etc...
* **Clear separation of symptoms and equivalent findings:** In scenarios where users are allowed to enter findings, it is important to differentiate the symptoms reported by the user from findings assessed by doctors. For instance a "skin rash" reported by a user is less reliable than the same finding reported by a trained dermatologist who can differentiate a "skin rash" from other similar symptoms.
* **Attribute support:** The symptom ontology should provide the means to describe all relevant details of a symptom presentation. For instance, if a symptom like vertigo (the sensation of movement that is not happening) is present for the differentiation of the corresponding vertigo diseases, it is important to collect attributes like duration, frequency, triggers, time since onset, exacerbating factors, relieving factors, and quality of perceived movement, amongst others. This also includes the definition of the scale types, ranges, states, and units of the attribute.
* **No redundancy and no overlap:** The symptom ontology should be free of redundancy. For example, there should not be several fever symptoms from merging several ontologies. There should also be no overlap e.g. not SymptomA, SymptomB and at the same time SymptomAOrSymptomB, SymptomAAndSymptomB etc... An explicitly defined symptom hierarchy, grouping symptoms from a users point of view may be optimal (in contrast to a hierarchy driven be medical terminology or pathophysiological considerations).

In addition to the symptom ontology, a single ontology is needed for all DSAA outputs that should be benchmarked. The most important of them is an ontology that can be used to describe the expected conditions. Selection criteria may include:

* **Common conditions:** The ontology needs to contain the all common conditions like common cold, viral sinusitis, etc...
* **Rare conditions:** The condition ontology should contain all rare diseases that otherwise might only be contained in special rare disease databases.
* **Idiopathic symptoms:** The ontology should support expressing idiopathic symptoms (without an underlying disease) even if they are not considered real diseases. In fact many users of DSAAs do not have a disease.

An ontology for pre-clinical triage may also be required. While triage in clinical settings is standardized by the Manchester-Triage-System and there are also clear rules for calling ambulances, there is no clear standard when to suggest traige levels such as "see a doctor tomorrow" or "see a doctor in 2-3 days".

A key criterion shared by all ontologies used is if they are maintained and updated on a regular base. There is the tendency that large, well maintained ontologies are quite noisy and less suited for DSAA, while the smaller ones are carefully curated, less noisy but also less complete. Another important dimension is the licence under which the ontologies are available. For instance for non member countries using Snomed CT is commercially not a viable option.

## Test Case Format

Given the numerous different details and the nested structure of symptoms for standardization a data format for description reference cases is needed. The search for such a format might include a thorough analysis of all available electronic health record formats. Given that the standardization might develop to cover more and more aspects and considering that case data already used for benchmarkings might be available for testing and training purpose, the ITU might decide to not rely on an external format but define its own. Given the flexible structure and medical doctors will be involved in defining an reviewing cases, dedicated domain specific language (DSL) might be considered too.

# Test Set Creation

Given the requirements it becomes clear that the creation of the test set is the most challenging part of DSAA benchmarking. While many clinics have large repositories with case data, they contain often only semistructured free text, the findings are not mapped to ontologies and the conditions are heavily biased by what the health insurance will pay for. This also affects the symptoms and findings that are often "optimized" to support the conditions that get the most money. The second large case source are the DSAA providers. However their cases are biased towards the questions they ask and also know to the DSAA provider that then could not participate. All cases from literature and journals can also not be used since the DSAA providers have probably already added them to their training sets or test sets.

Given there would be cases from a reliable source, they need to be labeled with expected outputs for all the features that should be benchmarked. The only source for labels are medical doctors. However, from experience is clear that the performance of doctors on labeling case data shows high variance. While some more senior doctors and experts perform quite well in their fields other do not. Therefore doctors might need to pass a test on known cases before they can be allowed for labeling. Even then, always several doctors should label the same cases. Possibly form different institutions (so reduce systematic error) and different specialties.

Given the challenges with the test set creation, for staring the standardization of DSAAs it could be an option to create hand curated 50-60 cases as it has been done for instance by the BMJ for their articles on symptom-checkers. The results are clearly not representative enough to base important strategic decisions on, but it could provide a promising starting point for the initiative.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_