MADAD: A Readability Annotation Tool for Arabic Text

Nora Al-Twairesh[1], Abeer Al-Dayel[2], Hend Al-Khalifa[3], Maha Al-Yahya[4], Sinaa Alageel[5], Nora Abanmy[6] and Nouf Al-Shenaifi[7]

[1,2,3,4,7]College of Computer and Information Science and [5,6]College of Pharmacy
King Saud University
Riyadh, Saudi Arabia

{[1]twairesh|[2]aabeer|[3]hendk|[4]malyahya|[5]salageel|[6]nabanmy|[7]noalshenaifi}@ksu.edu.sa

Abstract

This paper introduces MADAD, a general-purpose annotation tool for Arabic text with focus on readability annotation. This tool will help in overcoming the problem of lack of Arabic readability training data by providing an online environment to collect readability assessments on various kinds of corpora. Also the tool supports a broad range of annotation tasks for various linguistic and semantic phenomena by allowing users to create their customized annotation schemes. MADAD is a web-based tool, accessible through any web browser; the main features that distinguish MADAD are its flexibility, portability, customizability and its bilingual interface (Arabic/English).

Keywords: Readability, annotation, Arabic NLP

1. Introduction

Corpus annotation is defined as "the practice of adding interpretative linguistic information to an electronic corpus" (Garside et al., 1997). It is considered as an added value to the raw corpus and a crucial contribution to it (Garside et al., 1997).

Several types of annotations exist in Natural Language Processing (NLP) field, these include: structural annotation, POS tagging, morphological annotation, syntactic annotation, semantic annotation, pragmatic annotation and stylistic annotation. Also new types of annotation are emerging as new research fields are evolving in NLP, such as sentiment annotation and readability annotation.

Readability is defined as the degree to which a text can be understood (Klare, 2000). Assessing text readability is a long-established problem which aims to grade the difficulty or the ease of the text. Determining readability level is an important measurement to specify the possible audiences of text materials and to evaluate the impact on the readers. One of the obstacles Arabic readability research faces is the lack of sufficiently large data sets for which annotators provide labels with sufficient readability assessments. The construction of a corpus, which can serve as a gold standard to test new readability prediction tools, is needed.

In an effort to address this necessity, we propose MADAD, a collaborative online tool to construct a corpus of readability assessments for the Arabic Language. We named the tool MADAD "مدد" which is an Arabic term that means adding or increasing to an entity, since the main process of this tool is enriching the text with additional context.

The readability assessment feature that MADAD offers is flexible in which the readability assessments could be carried out on sentence, phrase and paragraph level. Furthermore, MADAD has two methodologies to construct a corpus of readability assessments: pair-wise comparison and direct evaluation of text difficulty. Up to our knowledge, we are the first to provide the readability annotation feature as a collaborative online tool to help in constructing training corpora for different readability services. On the other hand, MADAD can also be used as a general-purpose annotation tool for Arabic text. Since the tool allows its users to propose their ad-hoc annotation schemes; there are no specific annotation fields hard-coded in the tool. This way the tool can serve existing NLP tasks, and also new emerging fields.

The rest of the paper is organized as follows: section 2 presents previous attempts for creating Arabic annotation tools then sheds the light on the term readability and its assessment. Section 3 describes the main functionality of MADAD with an overview of the MADAD architecture. Finally, section 4 concludes the paper with future remarks.

2. Related work

Several Arabic annotation tools exist in the literature, however, most of them - if not all- are designed for a specific NLP task. There are tools for semantic annotation e.g. (Saleh & Al-Khalifa, 2009) and (El-ghobashy et al., 2014), dialect annotation e.g. (Benajiba & Diab, 2010) and (Al-Shargi & Rambow, 2015), morphological, POS-Tags, phonetic, and semantic annotation e.g. (Attia et al., 2009) and Arabic error correction e.g. (Zaghouani et al., 2014).

Most NLP tasks need a corpus for training machine learning classifiers, the corpus has to be in machine-readable format i.e. it has to be annotated for the machine to understand it. Evaluating text readability is one of these tasks.

The need of automated text readability assessment has been stimulated by the massive online sources along with advance in information technology. Measuring the text readability is important to meet people's information

needs and to predict if the text material is designed well to target the intended audience.

In readability field, interest is defined as gauging the relation between the intended reader and the written material. Defining this relationship is beneficial to scholars for educational purpose and for practitioners to help in selecting appropriate reading materials (Klare, 2000).

Assessing text readability is a long-established problem which aims to grade the difficulty or the ease of the text by defining the features that affects the reader. Determining readability level is an important measurement to specify the possible audience of text materials and to evaluate the impact on the readers. The traditional way to measure the text readability is through the use of formulas. These formulas are mathematical equations that take into account the characteristics of the text like length of words to predict the level of reading ability needed to understand the text (to objectively measure the relative difficulty of texts) (Klare, 2000).

The main purpose of these formulas is to provide human raters a simple approximation of the difficulty of a given text. Flesch Reading Ease score is an example of this formula which uses average sentence length along with average word length in syllables to calculate the readability degree. These formulas have a notable flaw in the methodology that is used to calculate the readability score. They do not have enough features to calculate the readability score so that it is impractical to predict maximal accuracy. These traditional measures are simple but shallow, and to overcome this drawback, data-driven machine learning approach is used for more accurate and robust analysis of text difficulty.

One example of machine learning approach is the work done by (Collins-Thompson, 2014). In their approach they evaluated the readability of a dataset using a variety of linguistic features combined with a prediction model. To train and test the readability prediction model, a gold-standard training corpus was used. In the training corpus the text is assigned a readability level by expert human annotators.

The next section describes the main functionality of MADAD with an overview of its architecture.

3. MADAD architecture

Figure 1 shows the overall architecture of MADAD. It consists of three layers and offers two modes of annotation tasks, the first mode is readability annotation and the second mode is schema-oriented annotation. The layers are: (1) the Services layer, which includes the MADAD corpus storage service, and the MADAD annotation services in order to process and coordinate the annotation tasks. (2) The Executive layer, which implements authentication and role assignment. (3) The User interface layer, which is shown to the user based on the user role (Administrator, Annotation Manager and Annotator). The Administrator will have the ability to create new Annotation Manager and Annotator accounts and to monitor the overall functionality of the tool. The mangers are responsible for defining the annotation task and assigning annotators for the task. Annotator is a person responsible for labelling (annotating) the text based on the defined task. All functionalities of MADAD are accessible via a web browser; the next section explains the main MADAD functions.

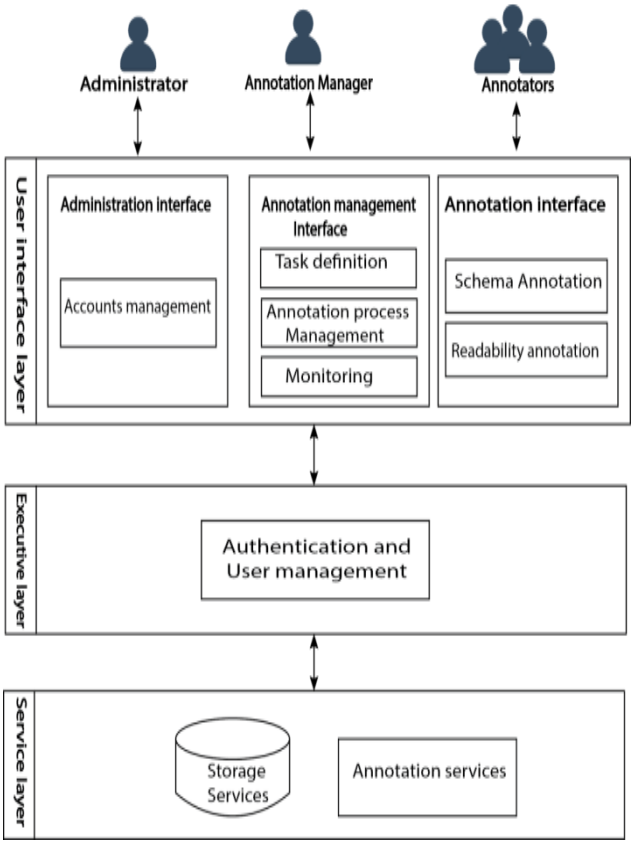

Figure 1: MADAD architecture

3.1 Creating corpus

In this function the task manger will import collection of text files with an option to pre-process the text by segmenting it into tokens and sentences. Madad offers two types of tokenization (sentence level and word level). In sentence level, the text will be splitted based on punctuation characters that defines the sentence boundaries e.g. semicolon and the full stop. In Word level, the text is segmented into atomic units (tokens) based on whitespaces.

3.2 Creating annotation task

The task Manager starts defining the annotation task by assigning value to these attributes (task name, description, annotation type annotation task guideline and number of annotators). Figure 2 shows a screenshot of create task.

Figure 2: Create task

MADAD offers two variations of methods to evaluate text readability level (comparison method and direct evaluation method). The comparison method provides pair wise comparison between two texts. Readability assessment through Comparison proven its value as a way to overcome the difficulty that the annotator may face to assign an absolute readability level for a given a text. In pairwise comparisons, it will be easy to judging two texts and decide which of them is more difficult (Tanaka-Ishii, K. et al., 2010).

If the task type is "direct readability evaluation method", then the Annotation Manager should define the scale range for the text difficulty. The default range is 0 (easy) to 100 (difficult). For the compression readability annotation mode, the task manager will define a set of comparisons statements for example "the text much easier", to determine the readability level between two texts.

To define the schema-orient annotation task, the manger will upload a schema file. Annotation schemas provide a means to define types of annotations. MADAD uses the XML Schema language supported by W3C [1] for the schema definitions. Figure 3 shows the schema structure. The main components of the schema are: Element declarations and Attribute declarations. The element declarations constrain the list of attributes the element can have. The attribute declarations define the values that the attribute may take.

Figure 4 shows example of annotation schema with element “date” and two attributes “year” and “time format” and list of assigned values for each attribute.

For each schema value the manager will assign a specific colour to facilitate the annotation task for the annotator by making it easy to distinguish each value from the rest of the schema values during the annotating text process (as shown in Figure 5).

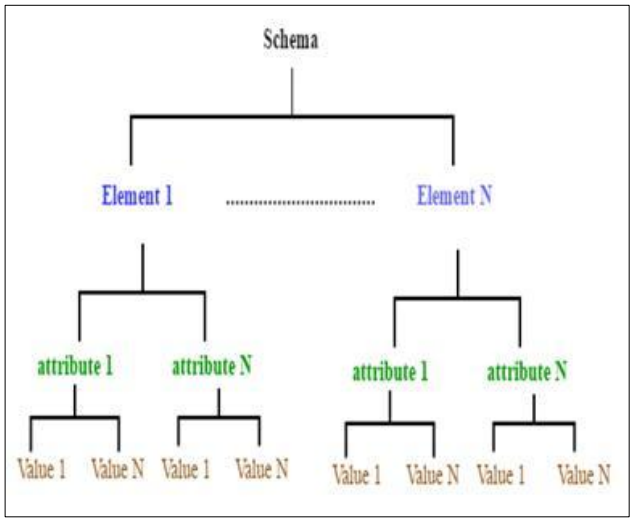

Figure 3: schema structure

```
<schema >
 <!-- XSchema deffinition for Date-->
  <element name="Date">
        <attribute name="year">
          <enumeration value="mm-yy"/>
          <enumeration value="dd-mm-yy"/>
        </attribute>
   <attribute name="time format">
          <enumeration value="24"/>
          <enumeration value="12"/>
     </attribute>

</element>
</schema>
```

Figure 4: Date schema

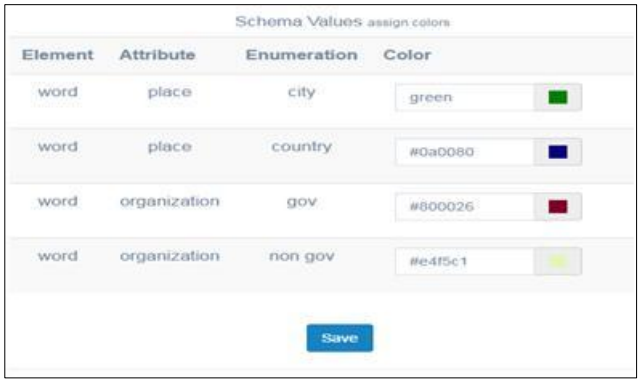

Figure 5: Assign color to schema values

3.3 Annotating text

The Annotator can view the assigned annotation tasks to start working on one of them. The tool will be flexible enough so that the Annotator can save any uncompleted tasks to pause the work on the annotation task and resume the work on the task later. Also the Annotator will be able to view the annotation guidelines with detailed descriptions of how to carry out the annotation task and how to treat different cases of annotation processes. The progress bar will be shown to illustrate the remaining texts in the corpus.

Based on the annotation type (Schema oriented or Readability annotation), the annotator will be able to (annotate) a text. The text will be shown to the annotator using file based representation.

[1] https://www.w3.org/XML/Schema

In the schema annotation task, the annotator will select some text in the page (based on the level of annotation, word or sentence) and assign value from predefined schema. The text will then be highlighted based on the value assigned colour.

The main functionality of readability comparison method is to allow the annotator to provide pairwise comparisons between the texts. In this method two texts will be displayed and the user is asked to compare between them based on previously defined statements like for example "the text is much easier". After selecting the comparison statement, a text pair and its corresponding assessment statement are added to the database and two new randomly selected texts appear to the annotator. For the direct readability evaluation method, the Annotator will be able to assign value to text based on the predefined range of text difficulty.

3.4 End annotation task

The task Manager will have the ability to end a running annotation task. After ending the task, two functions will be enabled to help the manager identify and reconcile differences in multiply annotated text (Evaluate annotation task and Adjudicating annotations). Multiple annotators will annotate the same text by using the same guidelines; to measure reliability of annotation the Inter-Annotator Agreement (IAA) will ensure that human annotator consistently makes same decisions based on the assumption that high reliability implies validity.

MADAD offers Evaluate annotation task function to help in assessin how well an annotation task is defined by using IAA scores. This function will provide a visual annotation comparison tool to see quickly where the differences are per annotation type.

If an IAA score is high, that is an indication that the task is well defined. This is typically defined using a statistical measure called a Kappa Statistic. For comparing two annotations results against each other, MADAD implements Cohen Kappa, however, in case of comparing more than two annotations results Fleiss Kappa will be applied (Bhowmick, et al., 2008).

In case that both annotators disagreed on annotating text, the manager will be able to resolve the conflict between the annotators using Adjudicating function.

In Adjudicating annotations function, the task manager will be able to edit and reconcile annotations manually. The task manager will compare the annotations values and determine which tags in the annotations are correct and should be included in the final version of the annotated corpus (gold standard).

In case of direct Readability evaluation mode and Readability comparison mode the IAA will calculate the expected chance agreement for each defined label.

In the schema oriented annotation task, the annotators have freedom to annotate any tokens in the text for that reason the IAA will calculate the frequency for the mutual annotated tokens. In that case the tokens' values that have been annotated by annotator1 and annottaor2 will be considered for agreement calculation.

3.5 Export annotated corpus

The task manager will be able to export the annotated corpus as xml file based on task definition. Table 1 illustrates the structure of the xml file. At the top of the table is the annotated text and token id underneath this appearing the annotations, one annotation per line. For each annotation its annotation level, Type, token Id, and Features is shown. The features are shown in the form "attribute = value".

Token ID	**Text with tokens**		
1	العربية		
2	يصبح		
Annotations			
Token Id	**annotation level**	**Type**	**Features**
1	word	Schema-orient	POS=Name
2	word	Schema-orient	POS=verb

Table 1: Result of annotations

4. Conclusion

In this paper we presented our approach for developing an Arabic readability annotation tool called MADAD, which offers an online tool to collect readability assessments for Arabic text. This tool will advance the research in the Arabic text readability field, by providing a method to construct a readability assessment corpus that serves as gold standard against which new readability scoring methods can be tested. Also, the tool provides schema-oriented annotation to be used in existing NLP tasks and new emerging tasks. This is done by giving the user the flexibility to define his/her own schema and not hard-coding the annotation tasks in the tool. This flexibility will increase the number of annotation tasks that potentially could use our tool.

MADAD also provides a user-friendly interface to serve different types of users from linguistic experts to novice users. In addition, it provides methods to evaluate different annotation tasks and gauging the agreements between annotators.

In the future, MADAD will be evaluated based on its ability to produce annotated corpus for readability annotation tasks and different NLP tasks. To gauge the effectiveness of the annotation process, we will compare MADAD with the available general purpose annotation tools according to an evaluation framework that is derived from (Dipper, et al., 2004) for annotation tools evaluation criteria.

Acknowledgment

This Project was funded by the National Plan for Science, Technology and Innovation (MAARIFAH), King Abdulaziz City for Science and Technology, Kingdom of Saudi Arabia, Award Number (INF 2822).

References

Al-Shargi, F., & Rambow, O. (2015). DIWAN: A Dialectal Word Annotation Tool for Arabic. In ANLP Workshop 2015 (p. 49).

Attia, M., Rashwan, M. A., & Al-Badrashiny, M. A. (2009). Fassieh, a semi-automatic visual interactive tool for morphological, PoS-Tags, phonetic, and semantic annotation of Arabic Text Corpora. Audio, Speech, and Language Processing, IEEE Transactions on, 17(5), 916–925.

Benajiba, Y., & Diab, M. (2010). A web application for dialectal arabic text annotation. In Proceedings of the LREC Workshop for Language Resources (LRs) and Human Language Technologies (HLT) for Semitic Languages: Status, Updates, and Prospects.

Bhowmick, P. K., Mitra, P., & Basu, A. (2008). An agreement measure for determining inter-annotator reliability of human judgements on affective text. In Proceedings of the Workshop on Human Judgements in Computational Linguistics (pp. 58–65). Association for Computational Linguistics.

Collins-Thompson, K. (2014). Computational assessment of text readability: A survey of current and future research. International Journal of Applied Linguistics, 165(2), 97–135.

Dipper, S., Götze, M., & Stede, M. (2004). Simple annotation tools for complex annotation tasks: an evaluation. In Proceedings of the LREC Workshop on XML-based richly annotated corpora (pp. 54–62).

El-ghobashy, A. N., Attiya, G. M., & Kelash, H. M. (2014). A Proposed Framework for Arabic Semantic Annotation Tool. Int. J. Com. Dig. Sys, 3(1), 47–53.

Garside, R., Leech, G. N., & McEnery, T. (1997). Corpus annotation: linguistic information from computer text corpora. Taylor & Francis.

Klare, G. R. (2000). The measurement of readability: useful information for communicators. ACM Journal of Computer Documentation (JCD), 24(3), 107–121.

Saleh, L. M. B., & Al-Khalifa, H. S. (2009). AraTation: an Arabic semantic annotation tool. In Proceedings of the 11th International Conference on Information Integration and Web-based Applications & Services (pp. 447–451). ACM.

Tanaka-Ishii, K., Tezuka, S., & Terada, H. (2010). Sorting texts by readability. Computational Linguistics, 36(2), 203-227.

Zaghouani, W., Mohit, B., Habash, N., Obeid, O., Tomeh, N., Rozovskaya, A., … Oflazer, K. (2014). Large scale arabic error annotation: Guidelines and framework. In Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14), Reykjavik, Iceland.