Arabic Word-level Readability Visualization for Assisted Text Simplification

Reem Hazim, Hind Saddiki, Bashar Alhafni, Muhamed Al Khalil, Nizar Habash
Computational Approaches to Modeling Language Lab
New York University Abu Dhabi
{rh3015,hind.saddiki,alhafni,muhamed.alkhalil,nizar.habash}@nyu.edu

Abstract

This demo paper presents a Google Docs add-on for automatic Arabic word-level readability visualization. The add-on includes a lemmatization component that is connected to a five-level readability lexicon and Arabic WordNet-based substitution suggestions. The add-on can be used for assessing the reading difficulty of a text and identifying difficult words as part of the task of manual text simplification. We make our add-on and its code publicly available.[1,2]

1 Introduction

Models for automatic readability assessment and automatic text simplification are relevant to many natural language processing (NLP) tasks such as developing pedagogical language technologies that assist students with learning languages, or teachers with curriculum design and writing assessment, as well as personalized paraphrasing of NLP systems' outputs to target different users with different readability levels.

Developing robust models for readability assessment and simplification requires the creation of large-scale lexical and annotated resources for training and evaluation. For example, parallel texts with different paired readability levels can be used to train readability models as well as simplification models. Figure 1 presents an example from an Arabic novel paired with a simplified version targeting the fifth-grade readability level. Properly identifying which words and phrases need to be rewritten in a simplified manner for a specific target readability level and audience requires word-level readability annotation in a framework that enables easy editing of the original text, as well as easy checking on the updated text. To our knowledge, most of the available tools for readability assessment work on the document or the sentence levels.

[1] http://samer-addon.camel-lab.com/
[2] https://github.com/CAMeL-Lab/samer-add-on

Original Text

نحن يا سيدتي في زمن مضطرب لا يركد عجاجه، ولا تسكن سيوفه في أغمادها، بعد أن انحلّت أواصر بني العبّاس، وأصبحت دولتهم أشلاءً ممزقة، يفترسها كل مفترس، ويُغير عليها كلُّ واثب.

We live, my lady, in a troubled age whose dirt-laden air does not settle, nor do its swords rest in their scabbards, after the bonds of the house of Abbas dissolved, and their state became like torn limbs preyed upon by every predator and raided by every opportunist.

Simplified Text

نحن يا سيدتي في زمن صعب لا يهدأ غباره، ولا تسكن سيوفه في جيوبها، بعد أن تقطعت صلات بني العبّاس، وأصبحت دولتهم قطعا ممزقة، يفترسها كل مفترس، ويهجم عليها كلُّ عدو.

We live, my lady, in a difficult age whose dust does not clear and whose swords do not rest in their sheaths, after the joints of the house of Abbas were cut off, and their state became like torn body parts preyed upon by every predator and attacked by every enemy.

Figure 1: An example original sentence from the Arabic novel "The Knight of Bani Hamdan" (Al-Jarim, 1945). The red-marked words are all of readability level 4 and 5 (difficult) per Al Khalil et al. (2018)'s readability lexicon. The simplified text rewrites those words into lower (easier) levels (green-marked words). The English translations are a best attempt to convey the complexity level of the Arabic word choices to non-Arabic readers.

The system presented in this paper addresses this limitation by focusing on word-level readability visualization to assist human annotators working on identifying text readability levels and adjusting texts to simplify them in a controlled setting. This effort is part of a project on the Simplification of Arabic Masterpieces for Extensive Reading (SAMER) (Al Khalil et al., 2017, 2018, 2020; Jiang et al., 2020). The project goals include the creation of a lemma-based graded readability lexicon for Arabic and a corpus of parallel original and simplified texts from Arabic novels (such as those presented in Figure 1). The project plans to target two different simplified readability levels: Grades 4-5 (Level III) and Grades 6-8 (Level IV).

While our focus is on Arabic, a language with

Proceedings of the The 2022 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pages 242 - 249
December 7-11, 2022

limited annotated resources for text simplification, the components we developed can be easily extended to other languages. The demo system is a Google Docs add-on that includes morphological analysis and light disambiguation of Arabic text, visualization of word readability levels, and access to substitution options with their own explicit readability levels. We make our add-on and its code publicly available.[1,2]

Next, we present some relevant Arabic linguistic facts (§2), and discuss related work (§3). We then present our design and implementation decisions (§4). In §5 we discuss some examples and use cases.

2 Relevant Arabic Linguistic Facts

Modern Standard Arabic (MSA) poses many challenges for NLP tasks (Habash, 2010). Two in particular are directly relevant to the task at hand, and affect many of our design decisions: morphological richness and orthographic ambiguity.[3]

Morphological Richness Arabic employs a combination of templatic, affixational, and cliticization morphological operations to realize a large number of features such as gender, number, person, case, state, aspect, voice, and mood, in addition to a number of attachable pronominal, preposition and determiner clitics. This leads to a very large number of words to model. To address this aspect, we utilize a morphological analysis component that is optimized for efficient representation (Graff et al., 2009; Taji et al., 2018).

Orthographic Ambiguity Arabic is commonly written with optional diacritical marks – which are often omitted – leading to rampant ambiguity. Orthographic ambiguity and morphological richness interact heavily with each other. For example the word فردها *frdhA*[4] has four core lemmas (Jiang et al., 2020): the verbs فَرَّد *far∼ad* 'individualize, separate in units', and رَدّ *rad∼* 'answer, return'; and the nouns فَرْد *fard* 'individual, unit' and رَدّ *rad∼* 'response, return'.

[3]We do not handle dialectal variants in this effort, although we acknowledge that dialectal differences from MSA are an important factor in readability assessment, since MSA is not the native variant of Arabic learned at home (Ferguson, 1959; Holes, 2004; Carroll et al., 2017).

[4]Arabic HSB transliteration (Habash et al., 2007).

Level	Grade	Age	Examples
I	1	6	بَيْت، كَبير، أَكَلَ، عَلى house, big, to eat, on
II	2-3	7-8	ذَهَب، أُسْطُواني، خَدَعَ، إذا gold, cylindrical, to cheat, if
III	4-5	9-10	رِئة، مُعادَلة، مُوَحَّد، أَغْرى lung, equation, united, to entice
IV	6-8	11-14	اِقْتِصاد، طُمَأنينة، راقٍ، نَكَثَ economy, tranquility, sophisticated, to breach
V	9+	15 -	أَدَمة، مِطْياف، لَوْذَعي، شُعَبيّ epidermis, spectroscope, witty, bronchial

Table 1: The five readability levels, their grade equivalencies, and lemma and English gloss examples, abridged from Al Khalil et al. (2020).

3 Related Work

Readability Resources Text readability leveling is relevant to a wide range of NLP applications such as text simplification and automatic readability assessment. Most research on readability leveling has focused on English, leading to the development of many resources (Collins-Thompson and Callan, 2004; Pitler and Nenkova, 2008; Feng et al., 2010; Vajjala and Meurers, 2012; Xia et al., 2016; Nadeem and Ostendorf, 2018; Vajjala and Lučić, 2018; Deutsch et al., 2020; Lee et al., 2021).

Specifically for MSA, datasets and modeling approaches have been created and developed by leveraging text targeted towards L1 readers (native speakers) (Al-Khalifa and Al-Ajlan, 2010; Al Tamimi et al., 2014; El-Haj and Rayson, 2016; Khalil et al., 2018) and L2 learners (non-native speakers) (Forsyth, 2014; Saddiki et al., 2018). More recently, Al Khalil et al. (2020) developed a 26,578-lemma lexicon (later extended to over 40,000 lemmas) with a five-level readability scale. Examples of vocabulary from the different readability levels and their corresponding grades and ages are shown in Table 1. This lexicon anchors readability at the lemma representation of the words. We use this lexicon as our reference for readability levels.

Jiang et al. (2020) developed the online Readability Leveled Arabic Thesaurus interface that leverages Al Khalil et al. (2020)'s lexicon, and extends its coverage.[5] For a given user input word, this interface provides the word's possible lemmas,

[5]http://samer.camel-lab.com/

roots, English glosses, related Arabic words and phrases from the Arabic WordNet (Black et al., 2006), and readability on a five-level readability scale. We make use of many components of Jiang et al. (2020)'s interface in our add-on.

Readability Visualization To the best of our knowledge, there has not been much work on developing web-based visualization tools for word-level readability assessment, neither for Arabic nor for other languages. Most of the existing tools work on the document or the sentence levels. Such tools include Readable[6] and datayze's Readability Analyzer[7] for English, and the recently proposed FABRA for French (Wilkens et al., 2022).[8] The lack of word-level tools for Arabic has motivated us to create an easy-to-use Google Docs add-on for word-level readability visualization.

Arabic Morphological Analysis and Disambiguation There are a number of tools that support Arabic morphological analysis and disambiguation and specifically lemmatization (Pasha et al., 2014; Darwish and Mubarak, 2016; Obeid et al., 2020, 2022). Inspired by the JavaScript Chrome extension developed by Khalifa et al. (2016) to assist Arabic learners in understanding text written in MSA or dialectal Arabic (DA), we implement a version of the Buckwalter core morphological analysis algorithm (Buckwalter, 2002) in JavaScript as part of our add-on.

4 Design and Implementation

4.1 Design Considerations

We designed our interface with the following considerations in mind.

Openness and Ease-of-use The system needs to be powerful and provides additive or complementary value to existing text editors, so that simplifications and changes can be evaluated on the fly and with minimal overhead. This needs to be accomplished with minimal usability tradeoffs.

Handling Arabic Ambiguity and Rich Morphology The system needs to be able to analyze fully inflected words and relate them to their lemmas and part-of-speech (POS) tags. The lemmas and POS tags will be used to identify the readability levels from Al Khalil et al. (2020)'s lexicon and to link with the Arabic WordNet databases (Black et al., 2006). Additionally, the interface needs to provide the users with access to all the analyses of a given word.

Visualizing Readability The interface needs to provide summary readability statistics in word-token and word-type spaces over full documents or arbitrary text selections. It should highlight the words in context in a clear way to indicate intuitively which words are easier and which are harder. And finally, the interface needs to provide access to the readability levels of other unchosen analyses of any word.

Access to Word Substitutions The system should support the text simplification process by displaying suggestions for related words and phrases, e.g., synonyms, antonyms, hypernyms, and hyponyms, with different readability levels. We build on the work of Jiang et al. (2020) who used the Arabic Wordnet to accomplish the same.

Explicit/Implicit Word Readability Markup The system should allow the recording of explicit readability levels such that when the automatic processes make mistakes, users can overwrite them. We want those corrections and annotations to be persistent across different future versions of the analyzer and lexicon. At the same time, unnecessary over-specification can be distracting to the reader or annotator and should be minimized. The system should support the ability to import and export text files that could be marked for readability using external tools.

4.2 Implementation

Google Docs Add-on We opted to implement our interface as a Google Docs add-on, which allows us to use one of the world's most used editing frameworks, without sacrificing any of Google Docs' advantages such as multi-author editing and other familiar word-editing supports.

We implemented the tool's front-end in HTML, CSS and JavaScript. The back-end was implemented in JavaScript, and it also utilizes the Apps Script Document Service, which is a JavaScript API used to read and modify Google Docs programmatically.

Readability Analysis and Visualization The tool analyzes user input in four main steps that are summarized in Figure 2. First, in the back-end, the

[6] https://readable.com/
[7] https://datayze.com/readability-analyzer.php
[8] https://cental.uclouvain.be/fabra/

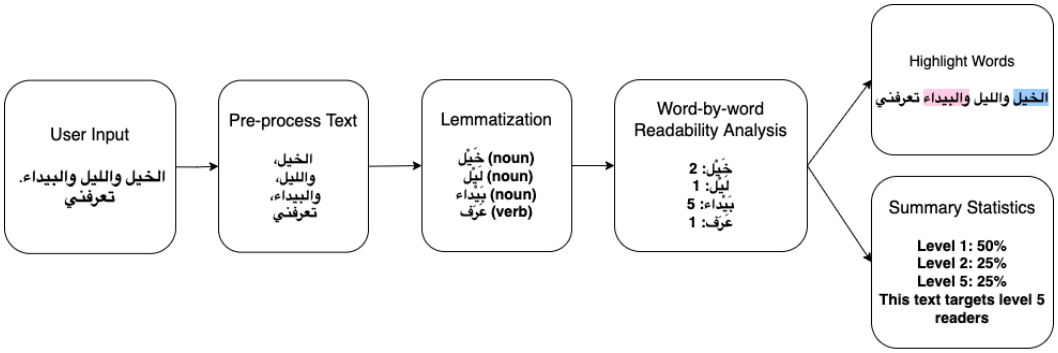

Figure 2: A flowchart depicting the steps that our tool takes to process user input. First, the user input is pre-processed and tokenized. Next, the lemma and part-of-speech of each token are determined using a morphological analyzer. Then, the tool looks up each lemma in the readability database to identify its readability level. The tool then highlights individual words accordingly and produces summary statistics describing the overall text readability.

text is pre-processed and tokenized, and non-word tokens are discarded. Second, the tokens are fed into the morphological analysis algorithm, which produces the most likely lemma and POS pair for each word. Third, we look up the lemmas in the readability database to identify their readability levels.[9] Finally, we use the Apps Script Document API to highlight words with different colors according to their readability levels. The tool also presents a summary of the text's readability distribution levels in a bar chart colored consistently with the readability level word highlights.

Morphological and Lexical Analyses Inspired by Khalifa et al. (2016)'s Chrome extension and Obeid et al. (2020)'s out-of-context MLE disambiguation mode, we implemented a version of the Buckwalter core morphological analysis algorithm (Buckwalter, 2002) in JavaScript as part of our add-on. Besides being used to determine readability levels, all lemma analyses are presented in a side bar to allow investigating and reassigning readability levels if needed. It is worth noting that the readability lexicon we use does not handle lexical polysemy. This is mainly due to the lexical representation that is used in the lexicon, which follows the representation of the Standard Arabic Morphological Analyzer (SAMA) (Graff et al., 2009). However, the design of our tool is independent of the granularity level of lexical representation and therefore, any updates to these components in the future can be easily integrated in our tool.

Figure 3 presents an instance of the SAMER Google Docs add-on with marked up text.

[9]We treat Proper nouns (Names) as a separate level.

Explicit/Implicit Word Readability Markup By default, the system deterministically specifies a readability level for any specific word based on its morphological and lexical readability resources. When disagreement with the automatic levels happen, as in automatic errors or importing text that was leveled externally, we ensure that the differences from the deterministic readability levels are not lost. To accomplish this, a prefix *#<i>#* is explicitly added in front of the word in question forcing the tool to interpret the word as having readability level of value *<i>*. For example, the word كتب *ktb* has a readability level of 1. However, the user can manually assign it a level of 5 by adding #٥# (Indo-Arabic digit *5*) in front of the word, like so: #٥#كتب *#5#ktb* . We also provide an interface button as part of the morphological side bar discussed above to make such assignment.

The add-on also provides multiple markup visualization modes to navigate between explicit and implicit readability level markup.

(a) **Show**: Explicitly mark all words with their readability levels.

(b) **Minimize**: Minimize all the markups by setting their font size to 1pt.

(c) **Hide**: Remove any markup whose readability level matches the internal level chosen by the analyzer and only keep the disagreeing markups. By default the Hide mode also minimizes the markup; however, the user can easily select the full text and resize it to a preferred font size (Hide+Resize).

(d) **Delete**: Delete all markup from the text.

Figure 4 shows the supported markup modes.

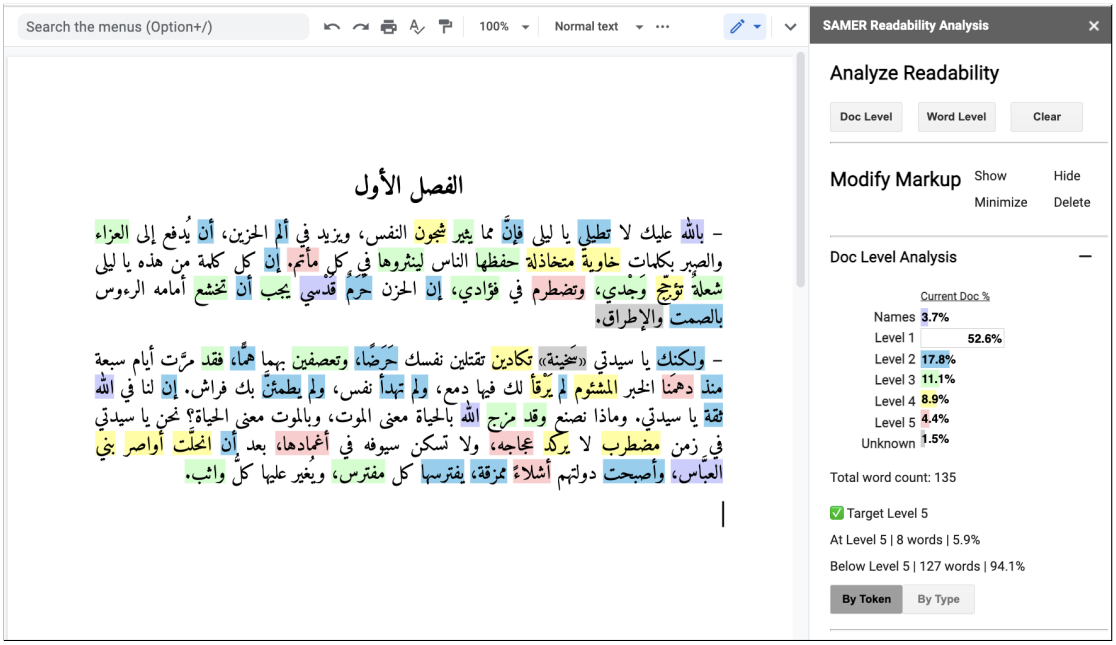

Figure 3: The SAMER Google Docs add-on visualizing word-level and document-level readability.

Markup	Example
Show	#١#علیها #١# كل #٣#مفترس، #٤#ويُغير #٢#يفترسها #٥#واثب.
Minimize	يفترسها كل مفترس، ويُغير عليها كلُّ واثب.
Hide	يفترسها كل مفترس، ويُغير عليها كلُّ واثب.
Hide + Resize	يفترسها كل مفترس، #٤#ويُغير عليها كلُّ #٥#واثب.
Delete	يفترسها كل مفترس، ويُغير عليها كلُّ واثب.

Figure 4: The different word-level markup modes that are supported by our tool.

5 SAMER Add-on: Examples and Use Cases

We present some examples of how the SAMER project Google Docs add-on can be used to analyze the readability of a literary text. We also discuss potential use cases of our tool across a variety of tasks and how it can be extended to other languages.

Examples Figure 3 shows the result of using the tool to analyze a short segment of a novel. After clicking on the `Doc Level` button at the top, the tool highlights each word according to its readability level using different colors, and presents a summary distribution of words in each readability level.

Figure 5 shows the result of selecting a specific word (انحلت *AnHlt* 'be disbanded') and clicking on the `Word Level` button at the top. A side bar appears showing the different lemma analyses by readability level. Various word substitution alternatives are presented to the user including synonyms, hypernyms and hyponyms, with their associated readability levels. If the user decides to change the word, they can simply rewrite it and rerun the readability analysis. If the user decides to change the automatically assigned readability level, they can either change it directly manually, or by clicking on the `Assign` button to change that specific word's readability level markup or the `Assign All` button to change all of its occurrences in the document.

Use Cases Our goal behind creating an easy-to-use Google Docs add-on tool for Arabic word-level readability analysis is to enable users to edit texts easily based on a specific target readability level. We intend for our tool to be used by human annotators to identify text readability levels and to simplify texts in a controlled setting. However, we envision that our tool can be used to assist writers in either making texts more sophisticated (harder readability) or in providing alternatives for specific words that have the same readability level.

Extending to Other Languages Although our work focuses on Arabic, the SAMER add-on tool is designed in a modular way and it can be easily extended to other languages. More concretely, the fol-

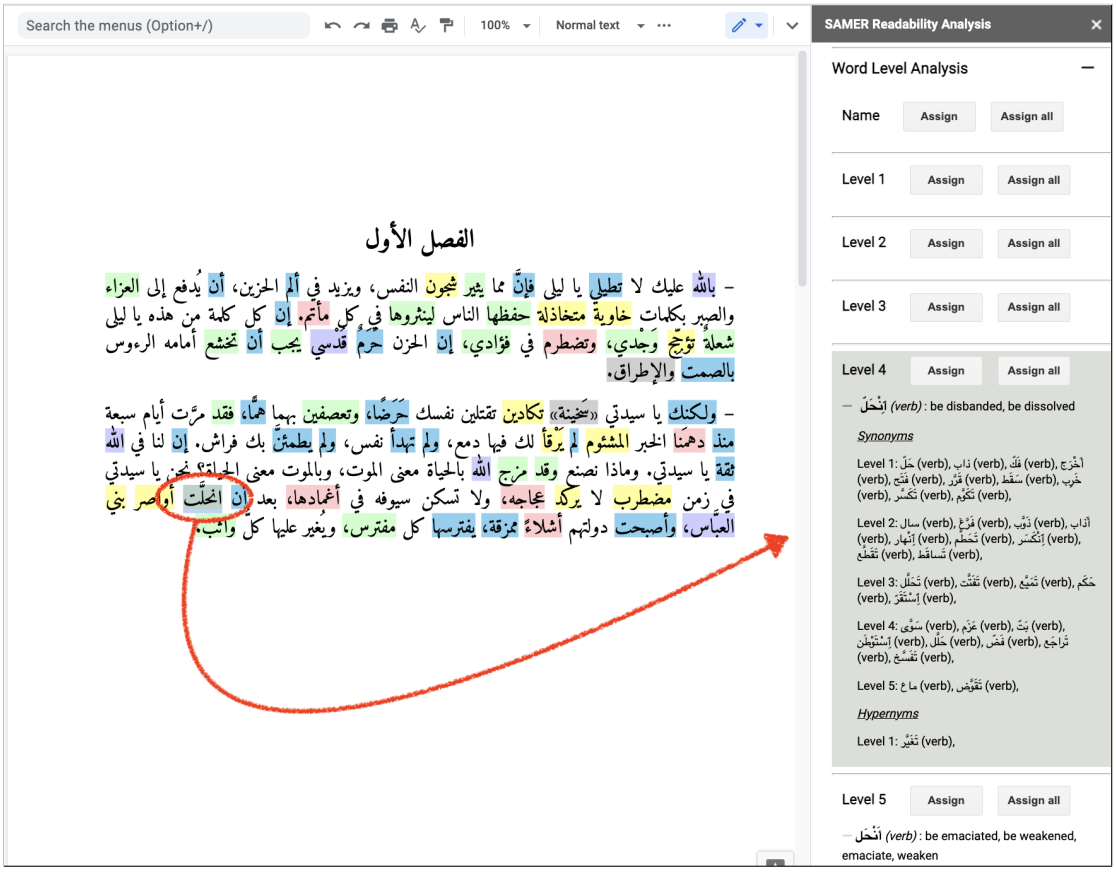

Figure 5: An example of selecting a specific word and identifying all of its analyses with their readability levels.

lowing core components are needed to make such an extension possible: (1) a readability level lexicon that relates lemmas to their readability levels; (2) a morphological analysis database that specifies prefixes, suffixes, stems and lemmas, and their co-occurrence compatibilities; (3) a statistical lemma-based disambiguation model; and (4) synonym, hypernym, hyponym and antonym lexical databases, such as those found in WordNet (Fellbaum, 2010).

6 Conclusion

We presented a Google Docs add-on for automatic Arabic word-level readability visualization. Our add-on includes a lemmatization component that is connected to a five-level readability lexicon and Arabic WordNet-based substitution suggestions. The add-on can be used for assessing the reading difficulty of a text and identifying difficult words as part of the task of manual text simplification.

In future work, we plan on enhancing our tool's readability analysis by leveraging additional morphosyntatic features (Saddiki et al., 2018). We will use the add-on to annotate a corpus of parallel original and simplified texts from Arabic novels.

Limitations and Ethical Considerations

We acknowledge that the add-on we developed could be used maliciously to: (a) modify texts under false pretenses, (b) plagiarize, or (c) profile people in a biased way using their writing style. We also acknowledge that automatic errors in readability analysis can lead to harmful results even when used with good intent. We further recognize that the use of highlighting as a visualization mechanism limits the conventional use of highlighting in text editing. Another limitation of our work is the lack of extrinsic and intrinsic evaluation. However, we are not aware of any manually annotated Arabic word-level readability datasets . We plan to develop such datasets using our tool. Finally, we acknowledge that further user studies are needed to confirm the effectiveness of our tool in aiding annotators to perform tasks such as text simplification.

Acknowledgements

This project is funded by a New York University Abu Dhabi Research Enhancement Fund grant. We thank Zhengyang Jiang, Go Inoue, and Ossama Obeid for helpful discussions.

References

Ali Al-Jarim. 1945. *Fāris Banī Ḥamdān (The Knight of Bani Hamdan)*. Hindawi Foundation.

Hend S Al-Khalifa and Amani A Al-Ajlan. 2010. Automatic readability measurements of the Arabic text: An exploratory study. *Arabian Journal for Science and Engineering*, 35(2 C):103–124.

Muhamed Al Khalil, Nizar Habash, and Zhengyang Jiang. 2020. A large-scale leveled readability lexicon for Standard Arabic. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 3053–3062, Marseille, France. European Language Resources Association.

Muhamed Al Khalil, Nizar Habash, and Hind Saddiki. 2017. Simplification of Arabic masterpieces for extensive reading: A project overview. *Procedia Computer Science*, 117:192–198.

Muhamed Al Khalil, Hind Saddiki, Nizar Habash, and Latifa Alfalasi. 2018. A leveled reading corpus of Modern Standard Arabic. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Abdel Karim Al Tamimi, Manar Jaradat, Nuha Al-Jarrah, and Sahar Ghanem. 2014. AARI: automatic Arabic readability index. *International Arab Journal of Information Technology*, 11(4):370–378.

William Black, Sabri Elkateb, Horacio Rodriguez, Musa Alkhalifa, Piek Vossen, Adam Pease, and Christiane Fellbaum. 2006. Introducing the Arabic wordnet project. In *Proceedings of the Global WordNet Conference (GWC)*, pages 295–300.

Tim Buckwalter. 2002. Buckwalter Arabic morphological analyzer version 1.0. Linguistic Data Consortium (LDC) catalog number LDC2002L49, ISBN 1-58563-257-0.

Kevin S Carroll, Bashar Al Kahwaji, and David Litz. 2017. Triglossia and promoting Arabic literacy in the United Arab Emirates. *Language, Culture and Curriculum*, 30(3):317–332.

Kevyn Collins-Thompson and James P. Callan. 2004. A language modeling approach to predicting reading difficulty. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*, pages 193–200, Boston, Massachusetts, USA. Association for Computational Linguistics.

Kareem Darwish and Hamdy Mubarak. 2016. Farasa: A new fast and accurate Arabic word segmenter. In *Proceedings of the Language Resources and Evaluation Conference (LREC)*, Portorož, Slovenia.

Tovly Deutsch, Masoud Jasbi, and Stuart Shieber. 2020. Linguistic features for readability assessment. In *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 1–17, Seattle, WA, USA → Online. Association for Computational Linguistics.

Mahmoud El-Haj and Paul Rayson. 2016. OSMAN — a novel Arabic readability metric. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 250–255, Portorož, Slovenia. European Language Resources Association (ELRA).

Christiane Fellbaum. 2010. Wordnet. In *Theory and Applications of Ontology: Computer Applications*, pages 231–243. Springer.

Lijun Feng, Martin Jansche, Matt Huenerfauth, and Noémie Elhadad. 2010. A comparison of features for automatic readability assessment. In *Coling 2010: Posters*, pages 276–284, Beijing, China. Coling 2010 Organizing Committee.

Charles F Ferguson. 1959. Diglossia. *Word*, 15(2):325–340.

Jonathan Forsyth. 2014. Automatic readability prediction for modern standard Arabic. In *Proceedings of the Workshop on Open-Source Arabic Corpora and Processing Tools (OSACT)*.

David Graff, Mohamed Maamouri, Basma Bouziri, Sondos Krouna, Seth Kulick, and Tim Buckwalter. 2009. Standard Arabic Morphological Analyzer (SAMA) Version 3.1. Linguistic Data Consortium LDC2009E73.

Nizar Habash, Abdelhadi Soudi, and Tim Buckwalter. 2007. On Arabic Transliteration. In A. van den Bosch and A. Soudi, editors, *Arabic Computational Morphology: Knowledge-based and Empirical Methods*, pages 15–22. Springer, Netherlands.

Nizar Y Habash. 2010. *Introduction to Arabic natural language processing*, volume 3. Morgan & Claypool Publishers.

Clive Holes. 2004. *Modern Arabic: Structures, Functions, and Varieties*. Georgetown Classics in Arabic Language and Linguistics. Georgetown University Press.

Zhengyang Jiang, Nizar Habash, and Muhamed Al Khalil. 2020. An online readability leveled Arabic thesaurus. In *Proceedings of the 28th International Conference on Computational Linguistics: System Demonstrations*, pages 59–63, Barcelona, Spain (Online). International Committee on Computational Linguistics (ICCL).

Salam Khalifa, Houda Bouamor, and Nizar Habash. 2016. DALILA: The dialectal Arabic linguistic learning assistant. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1098–1102, Portorož, Slovenia. European Language Resources Association (ELRA).

Muhamed Al Khalil, Hind Saddiki, Nizar Habash, and Latifa Alfalasi. 2018. A Leveled Reading Corpus of Modern Standard Arabic. In *Proceedings of the Language Resources and Evaluation Conference (LREC)*, Miyazaki, Japan.

Bruce W. Lee, Yoo Sung Jang, and Jason Lee. 2021. Pushing on text readability assessment: A transformer meets handcrafted linguistic features. In *Pro-*

ceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 10669–10686, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Farah Nadeem and Mari Ostendorf. 2018. Estimating linguistic complexity for science texts. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 45–55, New Orleans, Louisiana. Association for Computational Linguistics.

Ossama Obeid, Go Inoue, and Nizar Habash. 2022. Camelira: An Arabic Multi-Dialect Morphological Disambiguator. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2022) Demo Track*, Abu Dhabi, United Arab Emirates.

Ossama Obeid, Nasser Zalmout, Salam Khalifa, Dima Taji, Mai Oudah, Bashar Alhafni, Go Inoue, Fadhl Eryani, Alexander Erdmann, and Nizar Habash. 2020. CAMeL tools: An open source python toolkit for Arabic natural language processing. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 7022–7032, Marseille, France. European Language Resources Association.

Arfath Pasha, Mohamed Al-Badrashiny, Mona Diab, Ahmed El Kholy, Ramy Eskander, Nizar Habash, Manoj Pooleery, Owen Rambow, and Ryan Roth. 2014. Madamira: A fast, comprehensive tool for morphological analysis and disambiguation of Arabic. In *Proceedings of the Language Resources and Evaluation Conference (LREC)*, pages 1094–1101, Reykjavik, Iceland.

Emily Pitler and Ani Nenkova. 2008. Revisiting readability: A unified framework for predicting text quality. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 186–195, Honolulu, Hawaii. Association for Computational Linguistics.

Hind Saddiki, Nizar Habash, Violetta Cavalli-Sforza, and Muhamed Al Khalil. 2018. Feature optimization for predicting readability of Arabic L1 and L2. In *Proceedings of the 5th Workshop on Natural Language Processing Techniques for Educational Applications*, pages 20–29, Melbourne, Australia. Association for Computational Linguistics.

Dima Taji, Salam Khalifa, Ossama Obeid, Fadhl Eryani, and Nizar Habash. 2018. An Arabic Morphological Analyzer and Generator with Copious Features. In *Proceedings of the Fifteenth Workshop on Computational Research in Phonetics, Phonology, and Morphology (SIGMORPHON)*, pages 140–150.

Sowmya Vajjala and Ivana Lučić. 2018. OneStopEnglish corpus: A new corpus for automatic readability assessment and text simplification. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 297–304, New Orleans, Louisiana. Association for Computational Linguistics.

Sowmya Vajjala and Detmar Meurers. 2012. On improving the accuracy of readability classification using insights from second language acquisition. In *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP*, pages 163–173, Montréal, Canada. Association for Computational Linguistics.

Rodrigo Wilkens, David Alfter, Xiaoou Wang, Alice Pintard, AnaÃ¯s Tack, Kevin P. Yancey, and Thomas FranÃ§ois. 2022. Fabra: French aggregator-based readability assessment toolkit. In *Proceedings of the Language Resources and Evaluation Conference*, pages 1217–1233, Marseille, France. European Language Resources Association.

Menglin Xia, Ekaterina Kochmar, and Ted Briscoe. 2016. Text readability assessment for second language learners. In *Proceedings of the 11th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 12–22, San Diego, CA. Association for Computational Linguistics.