Camelira: An Arabic Multi-Dialect Morphological Disambiguator

Ossama Obeid[1], **Go Inoue**[1,2], **Nizar Habash**[1]
[1]Computational Approaches to Modeling Language (CAMeL) Lab
New York University Abu Dhabi
[2]Mohamed bin Zayed University of Artificial Intelligence
{oobeid,nizar.habash}@nyu.edu
go.inoue@mbzuai.ac.ae

Abstract

We present Camelira, a web-based Arabic multi-dialect morphological disambiguation tool that covers four major variants of Arabic: Modern Standard Arabic, Egyptian, Gulf, and Levantine. Camelira offers a user-friendly web interface that allows researchers and language learners to explore various linguistic information, such as part-of-speech, morphological features, and lemmas. Our system also provides an option to automatically choose an appropriate dialect-specific disambiguator based on the prediction of a dialect identification component. Camelira is publicly accessible at http://camelira.camel-lab.com.

1 Introduction

The last two decades have witnessed remarkable progress in Natural Language Processing (NLP) for Arabic and its dialects despite many challenges such as its diglossic nature, morphological complexity, and orthographic ambiguity (Darwish et al., 2021). These efforts have led to many practical applications for various NLP tasks including tokenization, part-of-speech (POS) tagging, morphological disambiguation, named entity recognition, dialect identification (DID), and sentiment analysis (Pasha et al., 2014; Abdelali et al., 2016; Obeid et al., 2019; Abdul-Mageed et al., 2020b, inter alia).

Tools for core technologies like POS tagging and morphological disambiguation are primary examples of such successful applications, e.g., MADAMIRA (Pasha et al., 2014), Farasa (Abdelali et al., 2016), UDPipe (Straka et al., 2016), and Stanza (Qi et al., 2020). However, there are still gaps to be filled in terms of coverage and usability. For example, these systems only support Modern Standard Arabic (MSA) and Egyptian Arabic, but not other widely spoken dialects such as Gulf and Levantine. In addition, these web interfaces only present the top prediction, although the alternative readings could provide valuable information for analyzing the models' behavior. In contrast, morphological analyzers such as ElixirFM (Smrž, 2007), CALIMA$_{Star}$ (Taji et al., 2018b), CALIMA Egyptian (Habash et al., 2012) show all the different readings for a given word out of context but without disambiguated analyses in context. These tools assume that users already know the input DID; however, this is not necessarily the case for second language learners.

To address these limitations, we present Camelira,[1,2] a web interface for Arabic multi-dialect morphological disambiguation that covers four major variants of Arabic: MSA, Egyptian, Gulf, and Levantine. Our system takes an input sentence and provides automatically disambiguated readings for each word in context, as well as its alternative out-of-context readings. We also showcase the integration of a state-of-the-art morphological disambiguator (Inoue et al., 2022) with the highest performing fine-grained Arabic DID system (Salameh et al., 2018) on the MADAR DID shared task (Bouamor et al., 2019). Camelira provides an option to automatically choose a dialect-specific disambiguator based on the prediction of the DID component. To the best of our knowledge, our work is the first to demonstrate an integrated web application that leverages both Arabic morphological disambiguation and DID systems.

Our contributions are as follows: (a) We present a user-friendly web interface that allows researchers and language learners to explore the detailed linguistic analysis of a given Arabic sentence. (b) We include three major Arabic dialects (Egyptian, Gulf, and Levantine) in addition to MSA, to make our tool more accessible to a wider audience. (c) We integrate DID to automatically select the appropriate disambiguator; a feature that helps users with limited knowledge of Arabic dialects.

[1]http://camelira.camel-lab.com

[2]Camelira is named after CAMeL Tools (Obeid et al., 2020), and in homage to MADAMIRA (Pasha et al., 2014).

Proceedings of the The 2022 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pages 319 - 326

2 Arabic Linguistic Facts

The Arabic language poses a number of challenges for NLP (Habash, 2010). We highlight three aspects that are most relevant to multi-dialectal morphological modeling: dialectal variations, morphological richness, and orthographic ambiguity.

First, Arabic is characterized with diglossia and its large number of dialects (Ferguson, 1959; Holes, 2004). MSA is the shared standard variant used in official contexts, while the dialects are the varieties of daily use. MSA and the dialects vary among themselves in different aspects, such as lexicons, morphology, and syntax. Second, Arabic is a morphologically rich and complex language. It employs a combination of templatic, affixational, and cliticization morphological operations to represent numerous grammatical features such as gender, number, person, case, state, mood, aspect, and voice, in addition to a number of attachable pronominal, preposition, and determiner clitics. Third, Arabic is orthographically highly ambiguous. This is due to its orthographic conventions where diacritical marks are often omitted, leading to a high degree of ambiguity. For example, MSA can have 12 different morphological analyses per word on average (Pasha et al., 2014).

3 Related Work

Morphological Analysis and Disambiguation Morphological analysis is the task of producing a complete list of readings (analyses) for a given word out of context. Morphological analysis has a wide range of applications, including treebank annotation (Maamouri et al., 2003, 2011, 2009) and improving morphological modeling (Habash et al., 2005; Inoue et al., 2017; Zalmout and Habash, 2017; Khalifa et al., 2020). Over the past two decades, there have been numerous efforts in building morphological analyzers for Arabic, e.g. BAMA (Buckwalter, 2002), MAGEAD (Habash and Rambow, 2006; Altantawy et al., 2010), ALMORGEANA (Habash, 2007), ElixirFM (Smrž, 2007), SAMA (Graff et al., 2009), CALIMA Egyptian (Habash et al., 2012), CALIMA Gulf (Khalifa et al., 2017), AlKhalil Morpho Sys (Boudlal et al., 2010; Boudchiche et al., 2017) and CALIMA$_{Star}$ (Taji et al., 2018b). Among these efforts, ElixirFM[3] and CALIMA$_{Star}$[4] provide easy-to-use web interfaces, allowing the user to explore all the possible morphological analyses for a given word. In addition to these rule-based approaches, Eskander et al. (2016) used a corpus-based paradigm completion technique (Eskander et al., 2013) to develop a morphological analyzer for Levantine; and (Khalifa et al., 2020) used the same technique to develop a morphological analyzer for Gulf.

Morphological disambiguation is the subsequent process of identifying the correct analysis in context from the list of different analyses produced by a morphological analyzer. Examples of this in Arabic start with MADA (Habash et al., 2005) and many following efforts (Pasha et al., 2014; Khalifa et al., 2016; Zalmout and Habash, 2017, 2020; Khalifa et al., 2020; Inoue et al., 2022), where they rank the analyses based on the predictions of morphological taggers. While these models have achieved significant improvement over time, only MADAMIRA (Pasha et al., 2014) offers a web interface[5] that's accessible to a general audience. In this work, we present a user-friendly web interface for state-of-the-art morphological disambiguation models to make these recent advances more accessible to a wider audience, such as linguists and language learners. Our interface also provides all the alternative readings of each input word with the associated prediction scores, allowing researchers to investigate the model's behavior.

Dialect Identification Dialect identification (DID) is the task of automatically identifying the language variety of a given text. DID for Arabic and its variants has attracted increasing attention in recent years. A number of shared tasks have been organized, including VarDial (Malmasi et al., 2016; Zampieri et al., 2017, 2018), MADAR (Bouamor et al., 2019), and NADI (Abdul-Mageed et al., 2020a, 2021, 2022), along with continuous efforts in dataset creation (Zaidan and Callison-Burch, 2011; Mubarak and Darwish, 2014; Zaghouani and Charfi, 2018; Baimukan et al., 2022, inter alia). These evaluation campaigns have led to the development of practical applications, such as ADIDA[6] (Obeid et al., 2019), a web interface for fine-grained Arabic DID based on the highest performing system in the MADAR shared task (Salameh et al., 2018). In this work, we employ one of the DID systems described by Salameh

[3] http://quest.ms.mff.cuni.cz/elixir
[4] http://calimastar.camel-lab.com/
[5] http://madamira.camel-lab.com/
[6] http://adida.camel-lab.com/

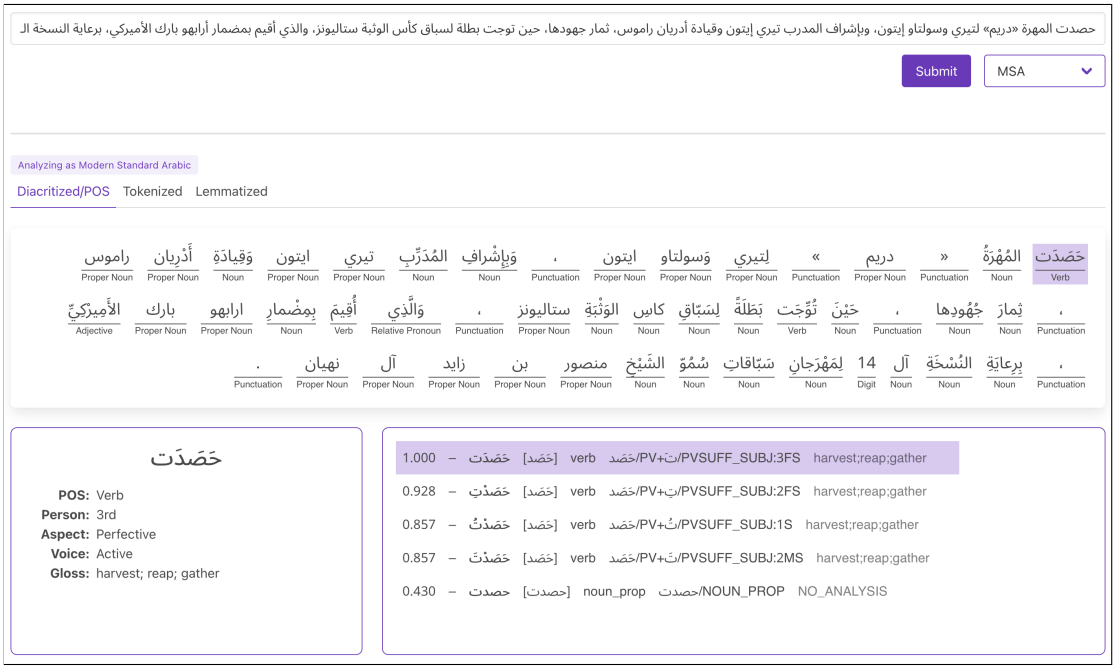

Figure 1: The Camelira interface with an MSA example sentence celebrating the winning of a racehorse named "Dream." In this example, the automatically diacritized forms of the words are presented together with their POS. The first word (on the right), which is highlighted, is selected by the user. The two lower boxes show all the possible out-of-context analyses (on the right) and the detailed features and gloss for the top in-context analysis (on the left).

et al. (2018);[7] however, we differ from their work in that we combine DID with multi-dialect morphological disambiguation to allow users to easily select an appropriate dialect-specific Arabic disambiguator based on the DID prediction.

4 System Design and Implementation

4.1 Design Considerations

We want an easy-to-use one-stop online-accessible user interface that supports the analysis of Arabic sentences from different dialects, and with access to under-the-hood decisions about disambiguation. To that end, we are inspired by three web interfaces: MADAMIRA (Pasha et al., 2014) for in-context disambiguation, CALIMA$_{Star}$ (Taji et al., 2018a) for out-of-context analysis, and ADIDA (Obeid et al., 2019) for dialect identification. Furthermore, we would like the web interface to have a responsive design with streamlined user experiences across a range of devices from mobile to desktops.

[7]We use regional level classification instead of fine-grained city-level classification because the morphological analyzers are designed at the regional level.

4.2 Implementation

Back-end The back-end is implemented in Python using `Flask`[8] to serve a REST API. We implemented the MODEL-6 DID system described by Salameh et al. (2018) for automatic dialect identification and the morphological disambiguation system described by Inoue et al. (2022). The implementation of the morphological disambiguator was provided by the CAMeL Tools[9] Python API (Obeid et al., 2020). We plan to add our MODEL-6 implementation to CAMeL Tools.

For morphological disambiguation, we use the *unfactored* model with a morphological analyzer for all variants. We chose the unfactored models because they are faster than the factored models and only slightly lower in performance. Table 1 shows the performance accuracy of Camelira's morphological disambiguation models. We report numbers on DEV as presented in Inoue et al. (2022).

For DID, we train our MODEL-6 using the TRAIN split and evaluate using the DEV and TEST splits following Salameh et al. (2018). Table 2 compares the performance of our implementation with that of Salameh et al. (2018). Our results are slightly lower due to implementation differences.

[8]https://flask.palletsprojects.com/
[9]https://github.com/CAMeL-Lab/camel_tools

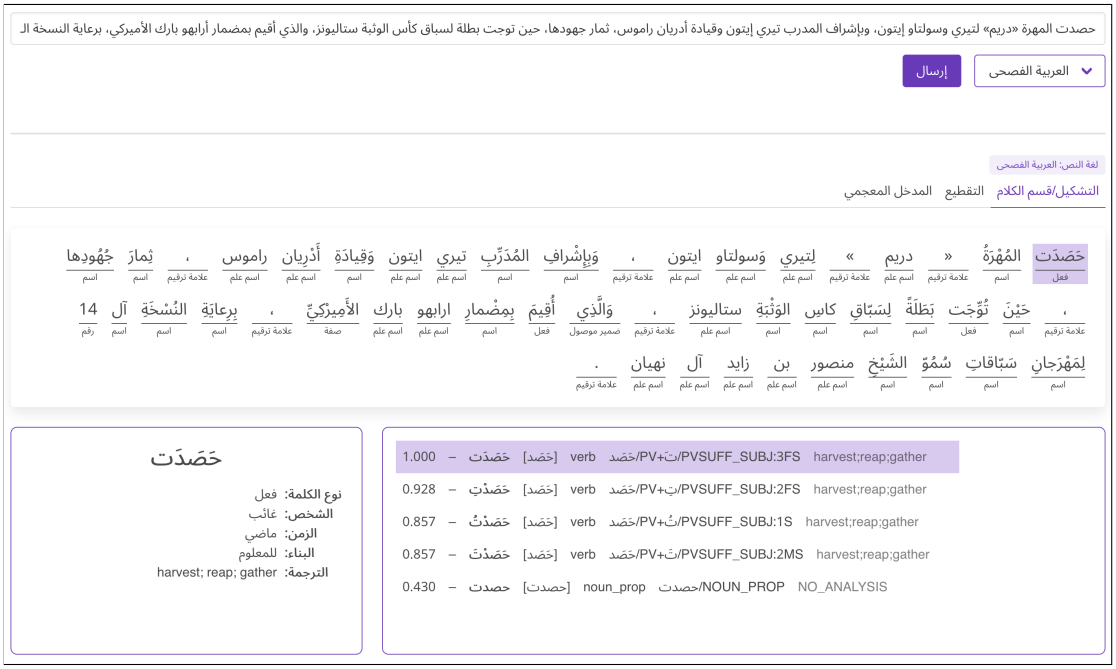

Figure 2: The Camelira interface presenting the same example in Figure 1 using the Arabic user interface.

	ALL TAGS	POS
MSA	95.9	98.7
EGY	90.5	94.0
GLF	93.8	96.6
LEV	85.5	92.7

Table 1: Accuracy of Camelira's morphological disambiguation models based on Inoue et al. (2022)'s unfactored+Morph models. **ALL TAGS** is the accuracy of the combined morphosyntactic features.

	DEV	TEST
Camelira	92.8	93.5
Salameh et al.	93.1	93.6

Table 2: Accuracy of Camelira's implementation of the MODEL-6 DID model compared with Salameh et al. (2018)'s implementation of the same model.

Front-end The front-end was implemented using `Vue.js`[10] for model view control and `Bulma`[11] for styling and creating a responsive design that works well across devices.

4.3 The Camelira Interface

The Camelira interface is divided into three main areas, the Input Area, Text Output Area, and Morphological Analysis Area. Figure 1 shows an example of a disambiguated MSA sentence in the Camelira web interface. We also provide the option of viewing the interface in Arabic as seen in Figure 2.

Input Area At first, only the Input Area is displayed which provides users with an input box where they can enter the sentence they wish to disambiguate. Users are also presented with a dropdown menu where they can select whether to disambiguate the input sentence as a particular dialect (MSA, Egyptian, Gulf, or Levantine) or to have the dialect be automatically selected.

Text Output Area Once the submit button is clicked and the sentence has been disambiguated, the Text Output Area is displayed. First, the dialect indicator displays which dialect was used to analyze the provided input. Then, an output box displays the disambiguated sentence in three different views: (a) the **Diacritized/POS** view which displays the diacritized text (if supported by the selected dialect's resources) along with the POS tag of each word, (b) the **Tokenized** view which displays each disambiguated word in its tokenized form where tokens are delimited by a '+' character, and (c) the **Lemmatized** view where each word is displayed in its lemmatized form. Figure 3 is the same as Figure 1 except that the text output is in Tokenized mode.

[10] https://vuejs.org/
[11] https://bulma.io/

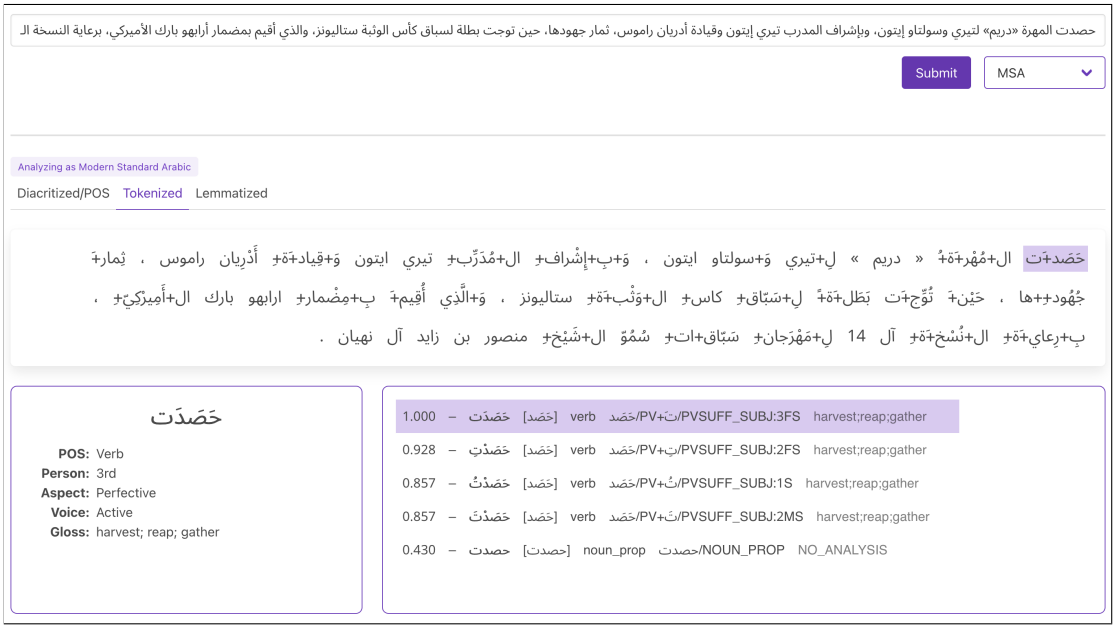

Figure 3: The Camelira interface with an MSA example sentence and "Tokenized" display tab. This is an exact replica of the input and output choices as in Figure 1 except that the word forms are presented in full tokenization.

Morphological Analysis Area Below the Text Output Area, the Morphological Analysis Area consists of the Analysis List box (on the right), which displays all analyses of a given word sorted by their disambiguation ranking order, and the Analysis Viewer box (on the left), which displays a selected analysis in an easy-to-read form with more morphological feature details. The analysis list displays the disambiguation score of each analysis as well as the values for a reduced set of features.

Clicking on a word in the Text Output Area selects that word, displaying its analyses in the analysis list and analysis viewer boxes. Clicking on an analysis in the Analysis List will display its user-friendly form in the Analysis Viewer. By default, the top analysis is selected.

Dialect Identification and Morphological Disambiguation Figures 4 and 5 present Egyptian and Gulf Arabic examples, respectively. Both are presented in a mobile setting to demonstrate our responsive design.

In the case of Figure 4, the user selected Auto-Detect for dialect identification. In the Gulf example, the user selected Gulf Arabic directly. Note that the Gulf Arabic does not show diacritizations since its training data did not include diacritized forms (Khalifa et al., 2020).

5 Conclusion and Future Work

We presented Camelira, a user-friendly web interface for Arabic multi-dialect morphological disambiguation that covers four major variants of Arabic. The system takes a sentence as input and provides an automatically disambiguated reading for each word, as well as its alternative readings, allowing users to explore various linguistic information, such as part-of-speech, morphological features, and lemmas. Camelira also provides an option to automatically choose an appropriate dialect-specific disambiguator based on the prediction of its dialect identification component.

In the future, we plan to extend our disambiguation system to cover other Arabic dialects such as Maghrebi and Yemeni Arabic. We also plan to continue to update the system using future improvements in terms of efficiency and accuracy in CAMeL Tools (Obeid et al., 2020).

Limitations and Ethical Considerations

We acknowledge that our system is currently limited to specific variants of Arabic and it can produce erroneous predictions especially on different dialects, genres, and styles that are not covered in the current system's training data. We also acknowledge that our work on core and generic NLP technologies can be used as part of the pipeline of other systems with malicious intents.

Figure 4: The Camelira interface with an Egyptian example sentence: *"A very cool song [video clip], you'll regret it if you don't watch it."* In this example, the input text is automatically correctly detected as Egyptian.

Figure 5: The Camelira interface with a Gulf example sentence: *"Go up to your room, I don't want to hear you talking about this subject again."* In this example, the user specified the input dialect as Gulf.

Acknowledgements

Some of this work was carried out on the High Performance Computing resources at New York University Abu Dhabi. We thank Salam Khalifa and Bashar Alhafni for their insightful comments and helpful discussions. We also thank anonymous reviewers for their helpful comments.

References

Ahmed Abdelali, Kareem Darwish, Nadir Durrani, and Hamdy Mubarak. 2016. Farasa: A fast and furious segmenter for Arabic. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 11–16, San Diego, California.

Muhammad Abdul-Mageed, Chiyu Zhang, Houda Bouamor, and Nizar Habash. 2020a. NADI 2020: The first nuanced Arabic dialect identification shared task. In *Proceedings of the Fifth Arabic Natural Language Processing Workshop*, pages 97–110, Barcelona, Spain (Online).

Muhammad Abdul-Mageed, Chiyu Zhang, AbdelRahim Elmadany, Houda Bouamor, and Nizar Habash. 2021. NADI 2021: The second nuanced Arabic dialect identification shared task. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 244–259, Kyiv, Ukraine (Virtual).

Muhammad Abdul-Mageed, Chiyu Zhang, AbdelRahim Elmadany, Houda Bouamor, and Nizar Habash. 2022. NADI 2022: The Third Nuanced Arabic Dialect Identification Shared Task. In *Proceedings of the Seven Arabic Natural Language Processing Workshop (WANLP 2022)*.

Muhammad Abdul-Mageed, Chiyu Zhang, Azadeh Hashemi, and El Moatez Billah Nagoudi. 2020b. AraNet: A deep learning toolkit for Arabic social media. In *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*, pages 16–23, Marseille, France. European Language Resource Association.

Mohamed Altantawy, Nizar Habash, Owen Rambow, and Ibrahim Saleh. 2010. Morphological Analysis and Generation of Arabic Nouns: A Morphemic Functional Approach. In *Proceedings of the Language Resources and Evaluation Conference (LREC)*, Valletta, Malta.

Nurpeiis Baimukan, Houda Bouamor, and Nizar Habash. 2022. Hierarchical aggregation of dialectal data for Arabic dialect identification. In *Proceedings of the Language Resources and Evaluation Conference*, pages 4586–4596, Marseille, France. European Language Resources Association.

Houda Bouamor, Sabit Hassan, and Nizar Habash. 2019.

The MADAR shared task on Arabic fine-grained dialect identification. In *Proceedings of the Fourth Arabic Natural Language Processing Workshop*, pages 199–207, Florence, Italy.

Mohamed Boudchiche, Azzeddine Mazroui, Mohamed Ould Abdallahi Ould Bebah, Abdelhak Lakhouaja, and Abderrahim Boudlal. 2017. AlKhalil Morpho Sys 2: A robust Arabic morpho-syntactic analyzer. *Journal of King Saud University - Computer and Information Sciences*, 29(2):141–146.

Abderrahim Boudlal, Abdelhak Lakhouaja, Azzeddine Mazroui, Abdelouafi Meziane, MOAO Bebah, and M Shoul. 2010. Alkhalil Morpho Sys1: A morphosyntactic analysis system for Arabic texts. In *Proceedings of the International Arab Conference on Information Technology*, pages 1–6.

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

Kareem Darwish, Nizar Habash, Mourad Abbas, Hend Al-Khalifa, Huseein T Al-Natsheh, Houda Bouamor, Karim Bouzoubaa, Violetta Cavalli-Sforza, Samhaa R El-Beltagy, Wassim El-Hajj, et al. 2021. A panoramic survey of natural language processing in the arab world. *Communications of the ACM*, 64(4):72–81.

Ramy Eskander, Nizar Habash, and Owen Rambow. 2013. Automatic extraction of morphological lexicons from morphologically annotated corpora. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1032–1043, Seattle, Washington, USA.

Ramy Eskander, Nizar Habash, Owen Rambow, and Arfath Pasha. 2016. Creating resources for Dialectal Arabic from a single annotation: A case study on Egyptian and Levantine. In *Proceedings of the International Conference on Computational Linguistics (COLING)*, pages 3455–3465, Osaka, Japan.

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

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. 2007. Arabic Morphological Representations for Machine Translation. In Antal van den Bosch and Abdelhadi Soudi, editors, *Arabic Computational Morphology: Knowledge-based and Empirical Methods*. Kluwer/Springer.

Nizar Habash, Ramy Eskander, and Abdelati Hawwari. 2012. A Morphological Analyzer for Egyptian Arabic. In *Proceedings of the Workshop of the Special Interest Group on Computational Morphology and Phonology (SIGMORPHON)*, pages 1–9, Montréal, Canada.

Nizar Habash and Owen Rambow. 2006. MAGEAD: A morphological analyzer and generator for the Arabic dialects. In *Proceedings of the International Conference on Computational Linguistics and the Conference of the Association for Computational Linguistics (COLING-ACL)*, pages 681–688, Sydney, Australia.

Nizar Habash, Owen Rambow, and George Kiraz. 2005. Morphological Analysis and Generation for Arabic Dialects. In *Proceedings of the Workshop on Computational Approaches to Semitic Languages (CASL)*, pages 17–24, Ann Arbor, Michigan.

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

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

Go Inoue, Salam Khalifa, and Nizar Habash. 2022. Morphosyntactic tagging with pre-trained language models for Arabic and its dialects. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1708–1719, Dublin, Ireland.

Go Inoue, Hiroyuki Shindo, and Yuji Matsumoto. 2017. Joint Prediction of Morphosyntactic Categories for Fine-Grained Arabic Part-of-Speech Tagging Exploiting Tag Dictionary Information. In *Proceedings of the Conference on Computational Natural Language Learning (CoNLL)*, pages 421–431, Vancouver, Canada.

Salam Khalifa, Sara Hassan, and Nizar Habash. 2017. A morphological analyzer for Gulf Arabic verbs. In *Proceedings of the Workshop for Arabic Natural Language Processing (WANLP)*, Valencia, Spain.

Salam Khalifa, Nasser Zalmout, and Nizar Habash. 2016. Yamama: Yet another multi-dialect Arabic morphological analyzer. In *Proceedings of the International Conference on Computational Linguistics (COLING)*, pages 223–227, Osaka, Japan.

Salam Khalifa, Nasser Zalmout, and Nizar Habash. 2020. Morphological analysis and disambiguation for Gulf Arabic: The interplay between resources and methods. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 3895–3904, Marseille, France. European Language Resources Association.

Mohamed Maamouri, Ann Bies, Hubert Jin, and Tim Buckwalter. 2003. Arabic treebank: Part 1 v 2.0. Linguistic Data Consortium (LDC2003T06).

Mohamed Maamouri, Ann Bies, Seth Kulick, Fatma Gaddeche, Wigdan Mekki, Sondos Krouna, and Basma Bouziri. 2009. The penn Arabic treebank part 3 v 3.1. Linguistic Data Consortium (LDC2008E22).

Mohamed Maamouri, Ann Bies, Seth Kulick, Fatma Gaddeche, Wigdan Mekki, Sondos Krouna, Basma Bouziri, and Wadji Zaghouani. 2011. Arabic treebank: Part 2 v 3.1.

Shervin Malmasi, Marcos Zampieri, Nikola Ljubešić, Preslav Nakov, Ahmed Ali, and Jörg Tiedemann. 2016. Discriminating between similar languages and Arabic dialect identification: A report on the third

DSL shared task. In *Proceedings of the Third Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial3)*, pages 1–14, Osaka, Japan. The COLING 2016 Organizing Committee.

Hamdy Mubarak and Kareem Darwish. 2014. Using Twitter to collect a multi-dialectal corpus of Arabic. In *Proceedings of the Workshop for Arabic Natural Language Processing (WANLP)*, Doha, Qatar.

Ossama Obeid, Mohammad Salameh, Houda Bouamor, and Nizar Habash. 2019. ADIDA: Automatic dialect identification for Arabic. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 6–11, Minneapolis, Minnesota.

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.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 101–108, Online.

Mohammad Salameh, Houda Bouamor, and Nizar Habash. 2018. Fine-grained Arabic dialect identification. In *Proceedings of the International Conference on Computational Linguistics (COLING)*, pages 1332–1344, Santa Fe, New Mexico, USA.

Otakar Smrž. 2007. ElixirFM — Implementation of Functional Arabic Morphology. In *Proceedings of the Workshop on Computational Approaches to Semitic Languages (CASL)*, pages 1–8, Prague, Czech Republic. ACL.

Milan Straka, Jan Hajič, and Jana Straková. 2016. UDPipe: Trainable pipeline for processing CoNLL-U files performing tokenization, morphological analysis, POS tagging and parsing. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 4290–4297, Portorož, Slovenia. European Language Resources Association (ELRA).

Dima Taji, Jamila El Gizuli, and Nizar Habash. 2018a. An Arabic dependency treebank in the travel domain. In *Proceedings of the Workshop on Open-Source Arabic Corpora and Processing Tools (OSACT)*, Miyazaki, Japan.

Dima Taji, Salam Khalifa, Ossama Obeid, Fadhl Eryani, and Nizar Habash. 2018b. An Arabic morphological analyzer and generator with copious features. In *Proceedings of the Fifteenth Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 140–150, Brussels, Belgium.

Wajdi Zaghouani and Anis Charfi. 2018. ArapTweet: A Large Multi-Dialect Twitter Corpus for Gender, Age and Language Variety Identification. In *Proceedings of the Language Resources and Evaluation Conference (LREC)*, Miyazaki, Japan.

Omar F Zaidan and Chris Callison-Burch. 2011. The Arabic Online Commentary Dataset: an Annotated Dataset of Informal Arabic With High Dialectal Content. In *Proceedings of the Conference of the Association for Computational Linguistics (ACL)*, pages 37–41.

Nasser Zalmout and Nizar Habash. 2017. Don't throw those morphological analyzers away just yet: Neural morphological disambiguation for Arabic. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 704–713, Copenhagen, Denmark.

Nasser Zalmout and Nizar Habash. 2020. Joint diacritization, lemmatization, normalization, and fine-grained morphological tagging. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8297–8307, Online.

Marcos Zampieri, Shervin Malmasi, Nikola Ljubešić, Preslav Nakov, Ahmed Ali, Jörg Tiedemann, Yves Scherrer, and Noëmi Aepli. 2017. Findings of the VarDial evaluation campaign 2017. In *Proceedings of the Fourth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial)*, pages 1–15, Valencia, Spain.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Ahmed Ali, Suwon Shon, James Glass, Yves Scherrer, Tanja Samardžić, Nikola Ljubešić, Jörg Tiedemann, Chris van der Lee, Stefan Grondelaers, Nelleke Oostdijk, Dirk Speelman, Antal van den Bosch, Ritesh Kumar, Bornini Lahiri, and Mayank Jain. 2018. Language identification and morphosyntactic tagging: The second VarDial evaluation campaign. In *Proceedings of the Fifth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2018)*, pages 1–17, Santa Fe, New Mexico, USA.