

Arabic NLG Language Functions

Wael Abed

Dept of Computing Science
University of Aberdeen
Aberdeen AB24 3UE, UK
w.abed.19@abdn.ac.uk

Ehud Reiter

Dept of Computing Science
University of Aberdeen
Aberdeen AB24 3UE, UK
ereiter@csd.abdn.ac.uk

Abstract

The Arabic language has very limited supports from NLG researchers. In this paper, we explain the challenges of the core grammar, provide a lexical resource, and implement the first language functions for the Arabic language. We did a human evaluation to evaluate our functions in generating sentences from the NADA Corpus.

1 Introduction

The Arabic Language is a member of the Semitic language family, the fifth most spoken language in the world. Around 330 million people speak this language as their mother tongue language distributed in 25 countries. It is mainly classified into two types; Classical and Modern Standard Arabic (MSA) (Al-Radaideh, 2020), in addition to regional variances (dialects). As far as we know, there are not any pipeline NLG systems or libraries that support an Arabic surface realiser at the time of writing this paper. Two factors might be the reasons: Arabic Language complexity and limited resources. An example that reflects the second cause is that even the Google n-gram viewer does not support the Arabic language (Alsmadi and Zarour, 2018).

Therefore, in this paper, we are presenting six simple morphological MSA natural language functions (inflectNoun, buildNP, inflectVerb, inflectAdjective, pronouns and numToWords). These functions can be considered as a good start to build an Arabic surface realiser, which can be used in a pipeline NLG system as they can be used to correctly generate and inflect different parts of speech including the generation of a noun phrase.

We implemented these functions from scratch using a rule-based approach. Furthermore, a lexicon containing around 9000 lemmata were collected. An open-source versions of lexicon and code with

a web interface are provided and available online on GitHub.¹

2 Related Works

As there is no NLG system done for Arabic, we investigated similar approaches that targets the language challenges. Green and DeNero (2012) presented a method to cover words agreements to generate correct words inflections in a translating task using a class-based agreement model. While, Khoufi et al. (2016) dealt with un-vocalisation and agglutination in Arabic. They built a statistical model to extract grammatical rules from a treebank corpus to use these rules for parsing.

Furthermore, another member of the Semitic group, which is the Hebrew language, also lacks NLG resources. One attempt to Generate a noun compounds in Hebrew goes back to 1998 (Netzer and Elhadad, 1998), they used a rule-based approach and listed the difficulties between English and Hebrew in terms of the syntactic, semantic and lexical constraints for generating a Noun Compounds, including a special feature of Hebrew called "Smixut".

For other languages, researchers used different realisers' technologies such as handcrafted grammars rule-based (Gatt and Reiter, 2009), statistical approaches (Gardent and Perez-Beltrachini, 2017) and human-generated template-based (Gatt and Krahmer, 2018). However, SimpleNLG is still considered the most popular method since it was presented by Gatt and Reiter (2009), It is a rule-based realiser that uses three main steps syntactic, morphological and orthographic realisations to produce a correctly inflected and high-quality output. It was adapted to many languages includ-

¹<https://github.com/WaelMohammedAbed/Natural-Language-Generation-for-the-Arabic-Language>

ing (in chronological order) French (Vaudry and Lapalme, 2013), Italian (Mazzei et al., 2016), Spanish (Ramos-Soto et al., 2017), Galician (Cascallar-Fuentes et al., 2018), Dutch (de Jong and Theune, 2018), Mandarin (Chen et al., 2018) and German (Braun et al., 2019).

Another popular approach is to provide grammars' functions to be used in a template-based NLG system. Such as RosaeNLG², which is an open-source, template-based, NLG library that gives simpler functions than SimpleNLG and had been used for English, French, German, Italian and Spanish languages. Similar approaches with fewer options can be found in Core-NLG³ and xSpin⁴.

Our approach is more similar to RosaeNLG than to SimpleNLG. As we will provide language functions and a noun phrase function to give correct grammatical inflections and agreements. Due to the huge morphological difference between the Semitic and non-Semitic languages, we could not adopt SimpleNLG nor RosaeNLG. Therefore, we found it convenient as a start to building our functions from scratch following the same approach of RosaeNLG.

3 Arabic language challenges

The Arabic language consists of 28 alphabet letters, written from right to left and there is no letters capitalisation. It has a very rich morphology measured in the number of different forms for its nouns, verbs and particles (Al-Radaideh, 2020).

3.1 Nouns

Arabic nouns can be inflected depending on the sentence gender, number, and grammatical case (nominative, genitive, and accusative).

Arabic assigns gender for living and non-living things (ex: "car" (سيارة) treated as feminine, while "Goal" (هدف) is a masculine noun) (Shamsan and Attayib, 2015). Moreover, unlike many languages, Arabic has a dual form in addition to singular and plural form. This form is used to express a noun that refers to two entities (things or people), and can be generated by adding one of these suffixes ("ان" - aan, "ين" - ayn, "تان" - taan, "تين" - tayn) to the singular form). The type of suffix used depends

²<https://rosaelng.org/rosaelng/1.16.4/about/compare.html>

³<https://github.com/societe-generale/core-nlg>

⁴<https://xspin.it/plus>

Singular feminine noun	سيارة (car)	
Dual and nominative	سيارتان	change the last letter (ة) to (ت) then add the suffix (ان) - aan
Plural and genitive	سيارات	remove the last letter (ة) then add the suffix (ات) - aat
Singular masculine noun	مدافع (defender)	
Dual and genitive	مدافعين	only add the suffix (ين) - ayn
Plural and nominative	مدافعون	only add the suffix (ون) - uun
Singular	Broken plural	Changes
باب - door	أبواب -doors	Add prefix "ah" (ا) and infix (و) (و)

Note: in the examples above there are other changes in the small vowels marks that did not mention because it will not be covered here

Table 1: examples show the Arabic noun's forms with different case and gender

on the case and the gender of the noun.

Regarding the plural form, there are two different types: sound plural and broken plural. The first one can be inflected using similar steps to the steps used in dual form, with a different set of suffixes ("ون" - uun, "ين" - iin, "ات" - aat). Some cases can only be identified by the "ḥarakāt" (حركات) short vowel marks of the last consonant letter. The short vowel marks will be neglected here, as for MSA according to many studies, adding these "Short Vowels Do Not Contribute to the Quality of Reading" (Abu-Rabia, 2019).

While the Broken plural has different inflections. This type has more than 30 patterns to use with internal and external manipulation of the sounds and vowels letters in the word and there are no firm rules (Dawdy-Hesterberg and Pierrehumbert, 2014). Table 1 illustrates all these states for dual, sound plural and broken plural.

3.2 Numerals

The challenges of the Arabic numerals are their relation with the noun's gender, case, definiteness, word order and counts. In addition, they do not follow the rules in some cases (Al-Bataineh and Branigan, 2020). These facts classify Arabic numerals as a complex problem compared to its relevance in the English language. Arabic numerals (ordinal and cardinal) can be classified mainly into three groups (Al-Bataineh and Branigan, 2020)(Al Easa,

Arabic example	Word by word to English	English translation
هدف واحد	Goal one(M)	one goal
سيارة واحدة	Car one(F)	one car
سيارتين اثنتين	cars(dual) two(F)	two cars
واحد وعشرون هدف	One(M) and twenty(m-plural) goal(m-singular)	Twenty one goals

Table 2: examples show the numerals inflection

1997).

Simplex numerals (1-10): according to (Al Easa, 1997) this group has two subgroups: singular numerals (1 and 2) and added numerals (3-10). Singular numerals agree with the counted noun in all aspects. However, this is the only group in the Arabic numerals that comes after the noun in the word order. For the second subgroup, the numeral gender will take the opposite form of the noun gender (see table 2).

Compound numerals (11-19): It has two major differences. The first one is that each numeral consists of two words, one for each digit (units and ten). For example, to say "twelve" in Arabic it will be "iṭnā 'ašara" (اثنا عشرة) which means "two" "ten". The second is the gender agreement. For "eleven" and "twelve", both parts of the numerals agree with the noun (Al Easa, 1997). While the rest from "thirteen" to "nineteen", the first part takes the inverse gender of the noun and the second part agrees (Al-Bataineh and Branigan, 2020).

Complex or doubled numerals (20-): One word is used for each digit and they connected by the conjunction "and" ("و" pronounced "wa"). The first digit (units) will follow the "Simplex numerals" rules (except for the word order). While the "tens" have one form which is the masculine plural form. The third digit "hundred" ("مئة" or "مائة" pronounced as "mi'a") considered as a feminine noun. The rest (fourth, fifth . . . etc.) digits treated as masculine nouns and use their plural form with numerals greater than two.

For ordinal numbers, the rules are mostly similar. However, there are a few differences. Such as ordinal numbers always agree with the noun gender, and nouns come before the numerals (Abdel-Malek et al., 1983).

3.3 Adjectives

In MSA, Adjectives commonly used after the nouns, although they can be used before as well. They agree with the noun in gender, case, definiteness and number (Shamsan and Attayib, 2015) (Idrissi et al., 2019). If the adjective is used to describe an un-human plural noun, then the adjective will take the singular feminine form (Shamsan and Attayib, 2015). However, with time this exception started to disappear and full agreement in the noun-adjective sentence is used (Idrissi et al., 2019).

Ex: الرجال الحكماء - the-men (M-plural-definite-nominative) the-wise (M-plural-definite-nominative)- The wise men

3.4 Verbs

Arabic verbs participate in a large percentage of the sentence due to their morphological richness. There are 19 forms for verb inflections, and for each form, the verb agrees with the subject in terms of number, person and gender. Moreover, it can be inflected using four moods, two voices and three tenses (Shalan et al., 2015).

3.5 Pronouns

Arabic pronouns can be categorised to Explicit and Implicit forms. The latter will not be discussed here as it only occurs in spoken Arabic. On the other hand, there are 3 branches can be derived from the Explicit form: attached, detached and reflexive (Igaab and Tarrad, 2019). Which can be used to create personal, demonstrative and relative pronouns (Ryding, 2005).

The attached (or dependent) pronouns are used as a suffix added to the verbs (ex: object pronouns), nouns (ex: possessive pronouns) and prepositions. In terms of the detached (independent) pronouns, they are used as a standalone word and inflected in more than one position in the sentence. Reflexive pronouns in Arabic is similar to that in English, the word "self" ("نفس", nafs) is used with the attached pronouns to form the reflexive pronouns. Such as ("نفسه") "nafsahu" (himself). All pronouns agree to what they refer to in person, gender and number.

4 Design and implementation of the Arabic language functions

In this section, we describe the Arabic lexicon, list the Arabic language functions proposed and what they cover from the challenges listed in section 3.

Type	1st person	2nd person	3rd person
m- singular-possessive	سي (ii) - my	ك (ka) - your	ه (hu) - his
m- plural-object	نا (naa) - our	كم (kum) - your	هم (hum) - their
m- singular-subject	انا (ana) - I	انت (anta) - you	هو (huwa) - he

Table 3: examples show the personal pronouns (attached and detached).

4.1 Arabic Lexicon

The lexicon was extracted from Wiktionary⁵. Usually, Wiktionary dump files are parsed to extract information (such as in (Braun et al., 2019)). However, the Arabic dump file is not well formatted. Therefore, a customised web scrap script generated to get all available nouns, verbs and adjectives from the English Wiktionary page. Overall, our lexicon covers around 9000 lemmata (~4000 nouns, ~3000 verbs and ~2000 adjectives).

4.2 Arabic language functions

To cover most of the challenges listed before, six main functions were designed (inflectNoun, buildNP, inflectVerb, inflectAdjective, pronouns and numToWords). Clauses and Pronouns types other than personal pronouns were not covered. The collected lexicon and collections of rules were used to implement these functions based on the information in section 3.

inflectNoun function: It gives the inflected form of the noun in the correct grammatical form. Taking into considerations all aspects number, gender and case. This is done by passing these values based on the sentence, then applying the rules to the targeted noun. The lexicon provides over 4000 nouns inflections, however, if the noun is not found it will be then inflected using an implemented collection of rules. Moreover, the dual and plural forms can be explicitly specified in the function call to get the desired result. (For all functions' definitions and parameters see the Appendices)

inflectAdjective function: Same as the noun it can return the right inflected form of the adjective based on the number, gender and case provided using the lexicon and rule-based method. Furthermore, it covers the case of a non-human noun and provides both full and deflected agreement.

⁵https://en.wiktionary.org/wiki/Category:Arabic_lemmas

numToWords function: It generates the respective words for a given number in two types "Cardinal" and "Ordinal". Such as 12 will spelt out as (اثنا عشر) twelve, if the "Cardinal" type is used. The generated text follows all the rules of the Arabic numerals mentioned in subsection 3.2.

buildNP (build Noun Phrase) function: It simply generated from the combining of the previous three functions. It will generate a noun phrase with numerals and a list of adjectives inflected correctly. It covers case, gender and number for nouns, numerals and adjectives in addition to the relationship and agreements between them.

inflectVerb function: It returns the inflected form of the verb after applying all the factors that can affect the verb form (number, gender, person, voice, mood and tense). The inflection first searched in the lexicon, if it is not provided then the verb will be inflected using a set of conditions, assuming that the verb belongs to the first form rules. This assumption was made because it is one of the most common forms (Shalan et al., 2015) and it takes a lot of time and effort to implements all forms for sound and weak verbs.

Pronouns function: This covers the personal pronouns. It can produce a correctly inflected pronouns for both attached and detached and for four types (subject, object, reflexive and possessive).

5 Evaluation

These functions are the first Arabic NLG functions and there are no other similar Arabic approaches to compare with. We evaluated our approach using the same two methods used in (Braun et al., 2019). The robustness and correctness of the implementation of the Arabic language functions were measured by examining 250 test units consist of different parts of speech (nouns, adjectives, verbs, numerals and pronouns). The results from the functions were all correct, whether it was inflected using a collections of rules or by using the lexical resource.

To measure how much these functions can cover from the Arabic language, NADA corpus is used (Alalyani and Marie-Sainte, 2018). The corpus categorised the data into ten categories. An average of 10 sentences extracted from each of the 10 categories provided in NADA and tried to generate them with our language functions. This was successful in 85% of cases.

Most of the sentences that the system failed to reproduce were because of the use of other pronouns types or wrong verb inflections when a verb is used in the sentence that is not a form (I) type and not in the lexicon. Such as this sentence "وأحرز كاريخو سيلفا هدف المباراة الوحيد" (Karejo Silva scored the game's only goal in the second minute), the verb "أحرز" (scored) is a singular, past, not form (I) and does not exist in Wiktionary. Another failed sentence is "وانتهت المباراة بفوز سندرلاند" (The match ended with the victory of Sunderland, who halted Liverpool's balance of 15 points), the word "الذي" (who) is a relative pronoun which can take different inflections. Therefore, this sentence considered as failed to reproduce, because relative pronouns are not yet covered by our functions.

As mentioned before, as far as we know this is the first NLG system for Arabic. Therefore, we could not compare it with others. However, for a similar approach with different system architecture and German language, Braun et al. (2019) achieved 84%, which means our results are acceptable. In terms of scalability, these functions can be adapted easily to use in Classical Arabic and all Arabic dialects (more than 20 dialects) Abdul-Mageed et al. (2018).

6 Future Work

We plan to enhance the performance of the system by covering other verbs forms, clauses and other types of pronouns (demonstrative and relative pronouns). Furthermore, we intend to increase the number of lemmata covered by the lexicon to get better accuracy, especially with weak verbs and broken plural cases. After that, we can create an Arabic version of simplenlg.

7 Acknowledgments

This work was carried out during the studying duration of the first author for an MSc of AI degree at the University of Aberdeen, which was fully funded by Chevening Scholarships, the UK government's global scholarship programme, funded by the Foreign and Commonwealth Office (FCO) and partner organisations. We also thank Jette Viethen and other people at Arria NLG for their suggestions and help.

References

- ZN Abdel-Malek, WM Erwin, and GN Saad. 1983. *Elementary Modern Standard Arabic: Volume 2, Lessons 31-45; Appendices*, volume 2. Cambridge University Press.
- Muhammad Abdul-Mageed, Hassan Alhuzali, and Mohamed Elaraby. 2018. You tweet what you speak: A city-level dataset of arabic dialects. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.
- Salim Abu-Rabia. 2019. The role of short vowels in reading arabic: A critical literature review. *Journal of psycholinguistic research*, 48(4):785–795.
- Hussein Al-Bataineh and Phil Branigan. 2020. The syntax of (complex) numerals in arabic. [Unpublished manuscript]. <https://doi.org/10.31219/osf.io/6pesa>.
- Noor Sultan Al Easa. 1997. Noun quantification in english and modern standard arabic: a descriptive morphological comparison. [Http://hdl.handle.net/10576/8069](http://hdl.handle.net/10576/8069).
- Qasem Al-Radaideh. 2020. Applications of mining arabic text: A review. In *Recent Trends in Computational Intelligence*. IntechOpen.
- Nada Alalyani and Souad Larabi Marie-Sainte. 2018. Nada: New arabic dataset for text classification. *International Journal of Advanced Computer Science and Applications*, 9(9).
- Izzat Alsmadi and Mohammad Zarour. 2018. Google n-gram viewer does not include arabic corpus! towards n-gram viewer for arabic corpus. *Int. Arab J. Inf. Technol.*, 15(5):785–794.
- Daniel Braun, Kira Klimt, Daniela Schneider, and Florian Matthes. 2019. Simplenlg-de: Adapting simplenlg 4 to german. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 415–420.
- Andrea Cascallar-Fuentes, Alejandro Ramos-Soto, and Alberto Bugarín Diz. 2018. Adapting simplenlg to galician language. In *Proceedings of the 11th International Conference on Natural Language Generation*, pages 67–72.
- Guanyi Chen, Kees Van Deemter, and Chenghua Lin. 2018. Simplenlg-zh: a linguistic realisation engine for mandarin. In *Proceedings of the 11th International Conference on Natural Language Generation*, pages 57–66.
- Lisa Garnand Dawdy-Hesterberg and Janet Breckenridge Pierrehumbert. 2014. Learnability and generalisation of arabic broken plural nouns. *Language, cognition and neuroscience*, 29(10):1268–1282.
- Claire Gardent and Laura Perez-Beltrachini. 2017. A statistical, grammar-based approach to microplanning. *Computational Linguistics*, 43(1):1–30.

Albert Gatt and Emiel Kraahmer. 2018. Survey of the state of the art in natural language generation: Core tasks, applications and evaluation. *Journal of Artificial Intelligence Research*, 61:65–170.

Albert Gatt and Ehud Reiter. 2009. Simplenlg: A realisation engine for practical applications. In *Proceedings of the 12th European Workshop on Natural Language Generation (ENLG 2009)*, pages 90–93.

Spence Green and John DeNero. 2012. A class-based agreement model for generating accurately inflected translations. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 146–155.

Ali Idrissi, Eiman Mustafawi, Tariq Khwaileh, and R Muralikrishnan. 2019. A neurophysiological study of noun-adjective agreement in arabic: The impact of animacy and diglossia on the dynamics of language processing. *bioRxiv*, page 729855.

Zainab Kadim Igaab and Intisar Raham Tarrad. 2019. Pronouns in english and arabic: a contrastive study. *Engl. Lang. Lit. Stud.*, 9(1).

Ruud de Jong and Mariët Theune. 2018. Going dutch: Creating simplenlg-nl. In *Proceedings of the 11th International Conference on Natural Language Generation*, pages 73–78.

Nabil Khoufi, Chafik Aloulou, and Lamia Hadrach Belguith. 2016. Parsing arabic using induced probabilistic context free grammar. *International Journal of Speech Technology*, 19(2):313–323.

Alessandro Mazzei, Cristina Battaglino, and Cristina Bosco. 2016. Simplenlg-it: adapting simplenlg to italian. In *Proceedings of the 9th international natural language generation conference*, pages 184–192.

Yael Netzer and Michael Elhadad. 1998. Generation of noun compounds in hebrew: Can syntactic knowledge be fully encapsulated? In *Natural Language Generation*.

Alejandro Ramos-Soto, Julio Janeiro-Gallardo, and Alberto Bugarín Diz. 2017. Adapting simplenlg to spanish. In *Proceedings of the 10th international conference on natural language generation*, pages 144–148.

Karin C Ryding. 2005. *A reference grammar of modern standard Arabic*. Cambridge university press.

Khaled Shaalan, Marwa Magdy, and Aly Fahmy. 2015. Analysis and feedback of erroneous arabic verbs. *Natural Language Engineering*, 21(2):271.

Muayad Abdul-Halim Ahmad Shamsan and Abdulmajeed Attayib. 2015. Inflectional morphology in arabic and english: a contrastive study. *International Journal of English Linguistics*, 5(2):139.

Pierre-Luc Vaudry and Guy Lapalme. 2013. Adapting simplenlg for bilingual english-french realisation. In *Proceedings of the 14th European Workshop on Natural Language Generation*, pages 183–187.

A Appendix 1 - Definitions and parameters of the Arabic language functions

1.1 ar_inflectVerb

Returns the inflected form of the input verb taking into consideration five aspects (number, gender, person, voice and mood/tense).

Parameter	Type	description
Word	string	(Mandatory) The input verb. Ideally, the verb root should be provided, however, if the word did not match any root in the DB, then, all other forms will be checked. If the word did not match again, this time the verb will be inflected manually using the known rules of the verb inflections
number	Number or String (Optional)	(default "plural") The grammatical number of the verb. Either singular, dual, plural or a number (not: dual or number 2 cannot be used with the 1st person)
Gender	Number or string (optional)	(default "1" for male). It accepts these values for male (1,m,male) and (0,f,female) for female.
Person	Number (optional)	(default "3" for the third person). The grammatical person of the verb. Either 1, 2 or 3 to indicate first, second and third person respectively.
Voice	String (optional)	(default "active"). It accepts only two values "active" and "passive" for the verb voice.
Mood_tense	String (optional)	(default "past"). It accepts five values ("past", "none_past", "subjunctive", "jussive" or "imperative"). However, if the value "passive" provided in the "Voice" parameter or 2nd person is selected then the value "imperative" is not acceptable.

Table 4: ar_inflectVerb parameters

1.2 ar_inflectNoun

Returns the inflected form of the input noun taking into consideration 3 aspects (number, gender and case).

Parameter	Type	description
Word	string (Mandatory)	The input noun. Ideally, the noun in its singular nominative form should be provided, however, if the word did not match any singular nominative form in the DB, then, all other forms will be checked. If the word did not match again, this time the noun will be inflected manually using the known rules of the noun inflections
number	Number or String (Optional)	(default "plural") The grammatical number of the verb. Either singular, dual, plural or a number (not: dual or number 2 cannot be used with the 1st person)
Gender	Number (optional)	(default "singular") The grammatical number of the noun. Either singular, dual, plural or a number.
Gender	Number or string (optional)	(default "1" for a male if the value, not in the DB). It accepts these values for male (1,m,male) and (0,f,female) for female
Case	String (optional)	The default value (nominative). This parameter accepts only these three values "nominative", "accusative" or "genitive". It represents the case of the noun to be inflected base on it.
Dual	String (optional)	If this value provided, then it overrides the existing "dual" value of the noun saved in the database. If not and there is no value in the database for the given noun then it will be treated according to the dual rules explained earlier.
Plural	String (optional)	If this value provided, then it overrides the existing "Plural" value of the noun saved in the database. If not and there is no value in the database for the given noun then it will be treated according to the Sound plural rules explained earlier.

Table 5: ar_inflectNoun parameters

1.3 ar_inflectAdjective

Returns the inflected form of the input adjective taking into consideration 3 aspects (number, gender and case).

Parameter	Type	description
Modifiers	List of Strings (Mandatory)	A list of adjectives that can be used to describe the noun given in the noun parameter. The full agreement between adjectives and the noun will be used, unless, the value of the "is_person" parameter is set to True and the value "DA" or False is given in the "agreement" parameter.
Number	Number or String (Optional)	(default "singular") The grammatical number of the noun. Either singular, dual, plural or a number.
Gender	Number or string (optional)	(default "1" for male). It accepts these values for male (1,m,male) and (0,f,female) for female. This value will be used in case of the adjective is not found in the database only.
Case	String (optional)	The default value (nominative). This parameter accepts only these three values "nominative", "accusative" or "genitive". It represents the case of the adjective to be inflected base on it
Is_person	Boolean (optional)	(default "True"). It indicates whether the modifier(s) refers to a person (human) noun or not.
Agreement	Boolean (optional)	(default "1", meaning Full agreement). This parameter now accepts these values (1, "fa", "full", "full agreement") for full agreement with the noun, and (0, "da", "deflected", "deflected agreement") for deflected agreement with the noun in case if the noun has the value "un-human" in the database.

Table 6: ar_inflectAdjective parameters

Parameter	Type	description
Number	Number or String (Optional)	(default "singular") The grammatical number of the pronoun. Either singular, dual, plural or a number (not: dual or number 2 cannot be used with the 1st person, therefore, in this case, the pronoun of the plural first person will be used)
Gender	Number or string (optional)	(default "1" for male). It accepts accepts these values for male (1,m,male) and (0,f,female) for female.
Person	Number (optional)	(default "3" for the third person). The grammatical person of the pronoun. Either 1, 2 or 3 to indicate first, second and third person respectively.
Type	String (optional)	(default "subject"). This value should be either subject, object, reflexive, possessive, or accusative object.
Word	string	(Optional if the type of the pronoun is independent and mandatory if not) (default Null), The input verb, noun or preposition. It will be used to add the pronoun as a suffix to it if the desired type of the pronoun is detached (dependent) pronoun
Plural	String (optional)	If this value provided, then it overrides the existing "Plural" value of the noun saved in the database. If not and there is no value in the database for the given noun then it will be treated according to the Sound plural rules explained earlier.
Is person	Boolean (optional)	(default "True") indicate if the word provided in the "word" parameter refers to a human or not. Because, for nonhuman plurals in the Arabic language, the singular feminine pronoun in its third-person form is used

Table 8: ar_pronouns parameters

1.4 ar_numToWords

Change the numbers to their respective words in two types "Cardinal" and "Ordinal". For example: 12 => (اثنا عشر) twelve, if the "Cardinal" type is used.

Parameter	Type	description
number	Number (Mandatory)	The number to be spelt out.
Case	String (optional)	The default value (nominative). This parameter accepts only these three values "nominative", "accusative" or "genitive". It represents the case that the number should be inflected base on it.
Gender	Number or string (optional)	(default "1" for male). It accepts these values for male (1,m,male) and (0,f,female) for female.
Type	String (optional)	Default (cardinal), it can take only two values: "cardinal" or "ordinal".
Number Format	Number or String (optional)	(default "wordsOnly"). If count is less than or equal to this value it will be spelt out; otherwise, it will be formatted as digits. For example, if you enter the integer 5, then numbers up to and including 5 will be words, and numbers greater than 5 will be digits. "wordsOnly", all numbers should be spelt out. "digitsOnly", no numbers should be spelt out.
Zero Format	String (optional)	(default "صفر"). The word used to express a value of 0. Either "صفر", "اي", or another word or phrase of your choice, e.g. "لا يوجد".

Table 7: ar_numToWords parameters

1.6 ar_buildNP

1.5 ar_pronouns

Returns the pronoun (e.g. you, his, her, them, etc.) for a given input gender or name. The optional parameters allow you to specify your desired pronoun type.

Generates a noun phrase with numerals in the right grammatical form, e.g. "مئتان و واحد و عشرون هدف" (Two hundred and twenty one goals).

Parameter	Type	description
Count	Number or string	(Mandatory) Either "dual", "plural" or a number. If a number is provided, it will be output as a modifier to the input noun and the noun is correctly inflected. If singular, dual or plural is used, the noun is inflected correctly based on the noun gender and the "case" parameter value.
Noun	String (Mandatory)	The input noun. Ideally, the noun in its singular nominative form should be provided, however, if the word did not match any singular nominative form in the DB, then, all other forms will be checked. If the word did not match again, this time the noun will be inflected manually using the known rules of the noun inflections.
Case	String (optional)	The default value (nominative). This parameter accepts only these three values "nominative", "accusative" or "genitive". It represents the case of the noun to be inflected base on it.
Gender	Number or string (optional)	(default "1" for male). It accepts these values for male (1,m,male) and (0,f,female) for female. If this value provided, then it overrides the existing gender of the noun saved in the database. If not and there is no value in the database for the given noun then it will be treated as "male" noun.
Dual	String (optional)	If this value provided, then it overrides the existing "dual" value of the noun saved in the database. If not and there is no value in the database for the given noun then it will be treated according to the dual rules explained earlier.
Plural	String (optional)	If this value provided, then it overrides the existing "Plural" value of the noun saved in the database. If not and there is no value in the database for the given noun then it will be treated according to the Sound plural rules explained earlier.
Number Format	Number or String (optional)	(default "wordsOnly"). If count is less than or equal to this value it will be spelt out; otherwise, it will be formatted as digits. For example, if you enter the integer 5, then numbers up to and including 5 will be words, and numbers greater than 5 will be digits. "wordsOnly", all numbers should be spelt out. "digitsOnly", no numbers should be spelt out
Zero Format	String (optional)	(default "صفر"). The word used to express a value of 0. Either "صفر", "اي", or another word or phrase of your choice, e.g. "لا يوجد".
Modifiers	List of Strings (Optional)	A list of adjectives that can be used to describe the noun given in the noun parameter. The full agreement between adjectives and the noun will be used, unless, the noun available in the database and has the value "un-human" and the value "DA" is given in the "agreement" parameter.
Agreement	Boolean (optional)	(default "1", meaning Full agreement). This parameter now accepts these values (1, "fa", "full", "full agreement") for full agreement with the noun, and (0, "da", "deflected", "deflected agreement") for deflected agreement with the noun in case if the noun has the value "un-human" in the database.

Table 9: ar_buildNP parameters