



UNIVERSITY OF LEEDS

This is a repository copy of *Computer-aided error annotation: a new tool for annotating Arabic error*.

White Rose Research Online URL for this paper:

<https://eprints.whiterose.ac.uk/83911/>

Conference or Workshop Item:

Alfaifi, AYG and Atwell, ES (2015) Computer-aided error annotation: a new tool for annotating Arabic error. In: 8th Saudi Students Conference, 31 Jan - 01 Feb 2015, Queen Elizabeth II Conference Centre, London.

Reuse

Items deposited in White Rose Research Online are protected by copyright, with all rights reserved unless indicated otherwise. They may be downloaded and/or printed for private study, or other acts as permitted by national copyright laws. The publisher or other rights holders may allow further reproduction and re-use of the full text version. This is indicated by the licence information on the White Rose Research Online record for the item.

Takedown

If you consider content in White Rose Research Online to be in breach of UK law, please notify us by emailing eprints@whiterose.ac.uk including the URL of the record and the reason for the withdrawal request.



eprints@whiterose.ac.uk
<https://eprints.whiterose.ac.uk/>



Computer-Aided Error Annotation

A New Tool for Annotating Arabic Error



Abdullah Alfaifi and Eric Atwell
University of Leeds {scaayga, e.s.atwell}@leeds.ac.uk



WHY THIS TOOL?

Existing tools for annotating errors in learner corpora are developed for languages other than Arabic. Thus, this poster introduces a new tool for computer-aided error annotation in Arabic learner corpora.

The new tool includes the Arabic Error Taxonomy (Alfaifi et al., 2013) which was designed specifically for Arabic

Error Category	Error Type	Tag
1. Orthography	1.1. Hamza (ء) error	<OH>
1. Orthography	1.2. Confusion in 'Ha' and 'Ta'	<OT>
	1.3. Confusion in 'Alif' and 'Ya'	<OA>
	1.4. Confusion in 'Alif' and 'Fatha'	<OW>
	1.5. Confusion between 'Nun' (ن) and 'Tawee' (ت)	<ON>
	1.6. Shortening the long vowels	<OS>
	1.7. Lengthening the short vowels	<OG>
	1.8. Wrong order of word characters	<OC>
	1.9. Replacement in word characters	<OR>
	1.10. Character(s) redundant	<OD>
	2. Morphology	2.1. Word inflection
2.2. Verb tense		<MT>
2.3. Other morphological errors		<MO>
3. Syntax	3.1. Case	<XC>
	3.2. Definiteness	<XF>
	3.3. Gender	<XG>
	3.4. Number (singular, dual and plural)	<XN>
	3.5. Word redundant	<XT>
	3.6. Word missing	<XM>
	3.7. Other syntactic errors	<XO>
4. Semantics	4.1. Word selection	<SW>
	4.2. Fatawa wasl (confusion in use of conjunctions)	<SF>
	4.3. Other semantic errors	<SO>
	4.4. Punctuation confusion	<PC>
	4.5. Punctuation redundant	<PT>
5. Punctuation	5.1. Punctuation confusion	<PC>
	5.2. Punctuation redundant	<PT>
	5.3. Punctuation missing	<PM>
6. Other	6.1. Other errors in punctuation	<PO>
	6.2. Other errors in punctuation	<PO>

Arabic Error Taxonomy

ARABIC LEARNER CORPUS

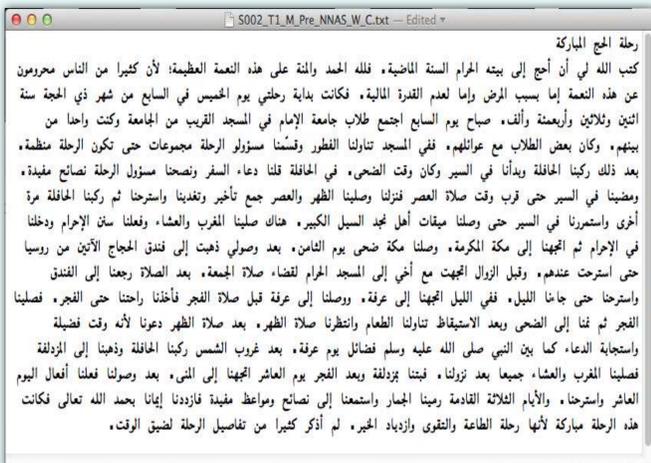
The error annotation will be applied to the Arabic Learner Corpus (ALC), it adds more value to the corpus data.

The corpus includes 282,732 words, produced by 942 learners of Arabic.



CURRENT DATA

The Arabic Learner Corpus currently includes 1585 texts in a raw format.



Example of raw text

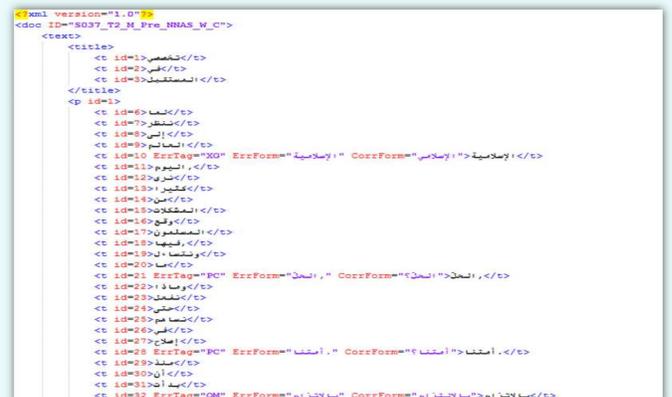
TARGET DATA

The ALC will include further information describing the language errors

Error Type Error Form Correct Form

<t id=10 ErrTag="XG" ErrForm="الإسلامية" CorrForm="الإسلامي">

Annotation



Example of annotated text

THE ANNOTATION TOOL

Main functions

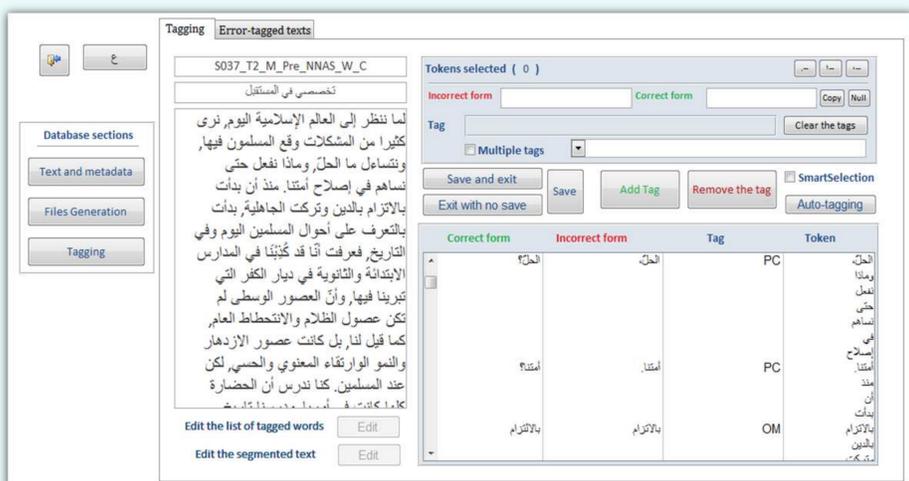
2. Smart-Selection

When an error exists more than once, the Smart-Selection function enables the annotator to find them all and tag them in a single step with no need to repeat the annotation process with each error.

3. Auto-Tagging

The Auto-Tagging feature, which is similar to translation memories, recognises the tokens that have been manually annotated and stores them in a database. So, tokens with the same error can be detected and annotated automatically next time.

For further information about ALC please visit: www.arabiclearnercorpus.com



1. Text Tokenisation

The Text Tokenisation function helps in segmenting the text into separate word-based tokens, this enables the annotator to attach tags to those tokens include errors.