Encodings, genres, texts: issues in Arabic corpus linguistics

Giuliano Lancioni[1]

1. Issues in the encoding of Arabic textual information

Computer encoding of information included in natural language texts is a complex task, which involves several distinct components. In this domain, as it happens in other human activities, we show a natural tendency to undervalue the complexities involved in realizing this task, since analyzing, and extracting information from texts is something we perform seamlessly and it is difficult to define consciously and accurately the steps needed to realize this process.

Texts themselves are in fact complex entities, which link together many pieces of information without many, if any, structured tags; in the most general case, ordinary written texts, only information tags can be found that are relative to boundaries at the general textual-rhetorical level, such as end of periods or sentences (marked by full stops or other larger pause markers – e.g., interrogation or question marks – with or without a newline), quotations (tagged by converging double quotes or isolated in indented blocks), parenthetical expressions (within parentheses or dashes or commas). In most cases, these pieces of information are relatively inconsistent or haphazard and often ambiguous: e.g., not all quotations are explicitly marked, and – as a growing body of literature is progressively showing – full stops are a highly ambiguous marker in natural language texts[2].

This state of affairs becomes even less clear-cut in Arabic texts, because of the general tendency in Arabic orthographic practices to an even larger inconsistence then can be found in most languages based on the Latin script, partly owing to a different tradition in the use of punctuation marks: Classical orthography in pre-modern age largely privileged the use of connectives rather than of explicit punctuation markers[3]. Markers in texts typically served to delimitate larger textual boundaries or to convey hints for the recitation. It should not be overlooked the fact that some of the graphical devices given for granted in European texts are simply not available in Arabic: e.g., uppercase letters, which are the standard devices to mark named entities in most European languages (with some excep-

1. Roma Tre Università.

2. On tokenization and sentence boundary detection, see Grefenstette and Tapaneinen, 1994.

3. For a resume of the general characters of Classical Arabic punctuation, the reader is referred to Gacek, 2009, p. 268-270, s. v. “Textual dividers and paragraph marks”.

tions, for instance uppercase used for all common nouns in German, or national adjectives in English) has no equivalent in the Arabic script.

More structured texts do exist, however: first of all, dictionaries: in a dictionary, even in relatively traditional ones, typographical devices – such as the use of different fonts (serif, sans serif, etc.), boldface, italics – or semi-structured markers (such as numbers and/or letters to distinguish different meanings of an entry or to separate homograph entries) are used to mark several kinds of lexicographic information: start of entries and subentries, phonetic realizations, morphology, definitions, homographs, distinct meaning of a word, and so on.

Arabic dictionaries can be singled out even in this case for some specificities: for instance, colors are used both in older editions of dictionaries and lexicons, as well as in some modern dictionary (such as *al-Munǧid*) in order to mark some specific pieces of information, typically entries within a root (which is the main level of organization in most Arabic dictionaries); or, as far as quotations are regarded, stars or other symbols for citation of poetic verses, special parentheses for Koranic quotations, and so on.

The revolution brought by the development of information technology, and in particular – as far as textual research is concerned – the emergence of contemporary markup languages (nowadays especially XML), forced researchers to rethink their methods and empirical practices. This revolution is, as often happens, twofold: encoders are given an enormous power to mark virtually every kind of information they deem relevant, without bothering for the usual empirical limitations (such as the impossibility to multiply typographical variants beyond a certain level without hindering readability); on the other hand, it is difficult to discipline this power, and encoders often end up with too large a list of tags, which makes it exceedingly difficult to yield comparable results from the same textual data by different annotators.

The only solution to this profligation of labels is to develop a reference framework for texts. Several projects have been developed, but the most successful one to date, at least as far as its acceptation as a standard is regarded is the Text Encoding Initiative (TEI). TEI predates XML, since it was created in 1994 as a consortium with the goal to define guidelines for the encoding of machine-readable texts. TEI guidelines P5, the last version of the standard released in 2007, amounts to a 1539-page book comprising an introduction, 23 chapters and 8 appendixes; most chapters of the guidelines are devoted to distinct textual genres and discuss specific issues in marking and tagging typical information for each genre.

The current version of TEI guidelines includes very large XML schema; the consortium provides also a software application, Roma, designed to create genre-specific subsets of the general TEI schema, which makes the system smaller and more manageable, as well as several TEI XSL transformations meant to produce output documents in several formats starting with a TEI-compliant document[4].

4. See Burnard and Bauman, 2009.

What distinguishes more clearly XML (and its incarnations such as TEI) from other encoding strategies is its strong orientation towards content, rather than representation. While in virtually all non-electronic texts, and also in many electronic ones such as traditional HTML files, we tend to consider representation as the same thing than content (which is in general not true), XML abstracts away content from its possible representations.

Separating content from its representation has advantages, but also disadvantages. On the one hand, it makes easier to develop information retrieval strategies in texts, since purely representational data are often irrelevant, when not outright damaging, for this kind of tasks.

TEI has been criticized for different reasons. The standard has been repeatedly been considered as too wide and too complex. While the full TEI standard is doubtless very large, it is difficult to conceive of an encoding which is able to find space for any reasonable piece of information in every possible textual genre while not requiring a large, and for that matter growing, set of tags. On the other hand, it is possible to encode in a TEI-compliant way by using only a subset of the full TEI set of tags, or "tagset"; in fact, the Roma application was devised exactly in order to generate useful, motivated subsets of the TEI tagset.

One more serious objection to the use of a general standard such as TEI is the difficulty to adapt its tags to the kinds of information that are specific of a textual subgenre. In the case of dictionaries, for example, the TEI tagset has often been regarded as too generic to represent the richness and diversity of information found in real-world lexicographic texts. However, this objection partly shows a couple of misconceptions on the role all-purpose textual markup systems such as TEI.

The first misconception is that every distinction found in a text should be reproduced as such by a markup language; in fact, only distinctions corresponding to functional differences should be represented, while merging should happen when two or more formats correspond to one and the same function. For instance, a text could use italics and double-quoted text to represent one and the same function (e.g., quotation; this might happen for lack of consistency, respect of tradition or every other reason): in this case, both conventions should clearly be represented by one and the same tag. In more complex cases, complete mismatches do happen between conventions and tags: encoding, exactly as textual criticism, should attain the goal to resolve typographic ambiguities, rather than to represent faithfully the letter of a text.

A second, perhaps subtler and more dangerous, misconception is that distinct tags should be available for every conceivable distinct piece of information in a text, while generic tags with distinct values for some attributes might do as well.

The main points in this discussion can be illustrated by an example from Arabic. A rather lesser-known Arabic-English dictionary (at least according to current Orientalist standards), Salmoné (1889), has been entirely digitalized ac-

cording to the TEI standard, in a single downloadable XML file, by the National Science Foundation at put in the public domain within the relatively small Arabic section of the Perseus Project[5].

Salmoné's electronic version choose exactly to translate relevant distinctions in the paper text by exploiting at length both TEI tags and attributes. For instance, in while two levels of organization – alphabetic letter and entry (with, possibly, a hierarchy of subentries or superentries) – are usually relevant in most European lexicographic works, the Arabic lexicographic tradition (see Haywood, 1965, for an introduction) has often a third, intermediate, level: the root. No tag is included in TEI's Dictionary section to directly encode roots; however, there exists a number of hierarchical tags (labeled <div1>, <div2>, and so on) aimed to encode increasingly subtler divisions of the text, which can be specialized through attributes.

This way, the electronic version of Salmoné's dictionary captures entry level distinctions by introducing a <div1> level with the type attribute set as "alphabetic letter" and a <div2> level with type "root" above ordinary entries.

2. A NEW "DISCIPLINE": COMPUTER AIDED ANNOTATION

Besides the definition of algorithms which are able to model reality (a goal often far from obtained in practice), one of the most important benefits computers have been able to produce in many different domains is the (semi-)automatization of a number of routine, time-consuming tasks, in order to let users concentrate on more strategic activities. The diffusion of this kind of computer-assisted activities gave rise to a series of computing domains generally labeled as "computer-aided", starting with the wide family of Computer-Aided Design (CAD) tools (perhaps the most successful example of this philosophy) and including Computer-Aided Manufacturing (CAM), Computer-Aided Composition (CAC), and so on.

In this context, I would like to introduce a new version of this idea, which might be call by analogy "Computer-Aided Annotation" (CAA). As the other examples above, CAA aims to give users, in this case annotators of textual data, specific tools which ease the task to annotate texts with relevant linguistic, rhetoric and lexicographic data. In order to do so, CAA tools should provide (semi-)automated information that is able to significantly reduce the time needed to annotate a text while limiting errors because of inconsistencies on the part of the annotator, distraction, mistyping, and so on.

Let us see some examples of how CAA might make easier and at the same time more robust the tasks involved in text annotation. One such task, an often complex and time-consuming one, is lemmatization: assigning every "word"

5. [http://www.perseus.tufts.edu/hopper/text?doc=Perseus%3Atext%3A2002.02.0005].
A new, larger effort to build a TEI XML version of a very large Arabic-English dictionary – in fact, the largest extant such a dictionary, namely *Edward William Lane's Arabic-English Lexicon*, 1863-1893 – has recently been put forth at the Perseus project with the joint funding of US Department of Education and the Max Planck Society.

– ranging, according to definition, from a graphical word separated by space, a morpheme, or even some more sophisticated linguistic entity – in a text to a lemma (again, however defined: from the naive concept of "dictionary entry" to linguistically more motivated definitions) with or without some additional information about morphology, syntax (e.g., valency) and/or phonology.

To be concrete, the Arabic word *katabtu*, "I wrote", might be lemmatized as *kataba*.1s.pf – "to write, 1st singular, perfect". This kind of annotation might assures many advantages: for example, one might search all occurrences of "to write" in a text, independent of the morphological form taken by the verb in a specific context (and, in a language such as Arabic, independent from attached elements such as bound pronouns, e.g. in *katabtuhu* "I wrote it"); or, alternatively, all contexts might be retrieved where perfect verbal forms are used in order to study their distribution.

Lemmatization is far from being a trivial task. Among the many problems it involves, the most striking is perhaps relative to consistency: as strange as it can seem, experience shows that it is unlikely that two annotators – and even the same annotator in different contexts – lemmatize in exactly the same way. Shared reference dictionaries and strict annotation guidelines may ease, but are not likely to eliminate, this problem[6].

CAA tools can help in this task in several ways. The simplest, yet very useful, aid is just to provide the annotator the list of lemmatization alternatives previously entered for each word: to stick to the previous example, if *katabtu* has already been entered before, "*kataba*.1s.pf" will appear and the annotator shall just have to confirm or to enter deliberately a new alternative; this will suffice to avoid some of the most frequent typical errors – e.g., marking "perfect" with another abbreviation rather than "pf", forgetting person of number marking, mistyping the lemma.

This type of lemmatization aid, while having the advantage of simplicity of implementation and use (just a simple dictionary of alternative has to be maintained), has the disadvantage that the building of a reference lemmatization dictionary from scratch is a slow process, and until a relatively late stage most words will have no existing lemmatization to be proposed. A stronger alternative, which we shall explore in next section, is to dwell upon a generative mechanism, or a 'lemmatizer', which is able to propose possible lemmatization alternatives for most words in a text, even if they have not been met with before.

A better solution might been a mix-up of these two strategies: lemmatization alternatives already entered will be presented first if available, perhaps marked in bold or highlighted in some other way, together with other choices propo-

6. One of the most thorough list of lemmatization recommendations and rule for Arabic can be found in Kaplony, 2011, within the Arabic Papyrology Database Project.

sed by the lemmatizer, sorted – if enough information is available – by decreasing frequency of lemmas and/or grammatical features[7].

The next section will review some of the CAA projects under way in the Roma Tre research group on Arabic linguistics.

3. Projects of the Roma Tre Arabic research group

Some CAA tools have been developed by the Roma Tre research group on Arabic linguistics around a central research project, SALAH – an acronym that stands for Segmentation And Linguistic Analysis of Ḥadīṯ texts. The idea behind SALAH is to develop a general model for the unsupervised segmentation and analysis of texts within a specific textual genre, in this case *ḥadīṯ*s, or Prophetic traditions[8].

The model currently provides an automatic segmentation of each Prophetic tradition in a chain of transmitters or *isnād* – a central issue in establishing the validity of *ḥadīṯ*s within the Islamic juridical disciplines – and the text proper, or *matn*, and further analyses each segment according to two distinct processes: on the one hand, a set of regular expressions chunks each transmitter chain in a graph labeled with the relations between transmitters, while a tailored, augmented version of the AraMorph morphological analyzer analyzes and annotates lexically and morphologically the text content.

The general schema of process and flows included in SALAH is depicted in Figure 1 :

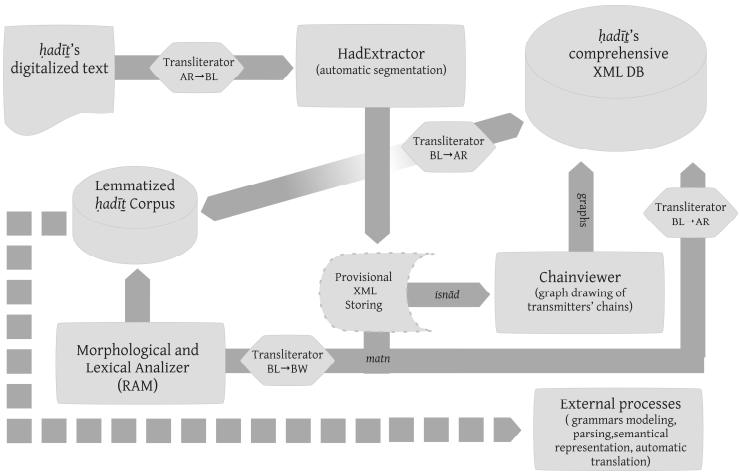

Figure 1. The SALAH process flowchart

7. A strong aid to this end might be provided by the recently published frequency dictionary by Buckwalter and Parkinson, 2011, once the printed material (including rank, frequency, lemmas, English translations and illustrative examples) becomes available in electronic format.

8. For a deeper description of the SALAH project, see Boella *et al.*, 2011.

The final output of the system is a graph containing relations among transmitters and a lemmatized text corpus, both in XML format. This output can further feed, for instance, the automatic generation of concordances of the texts with variable-sized windows and according to several possible formats (one advantage of the content-representation distinction that characterizes XML). The results can be useful for a variety of purposes, including retrieving information from *ḥadīṯ* texts, verify the relations between transmitters, finding variant readings, supplying lexical information to specialized dictionaries[9].

As an example, we shall delve somewhat deeper in the mechanism of the core lemmatizer of the SALAH project, the Revised AraMorph (RAM) module. The starting point for this lemmatizer is Tim Buckwalter's (2002) AraMorph (AM), whose outstanding features are its simplicity of design, its high performance even in unsupervised environment and the easiness of its maintaining and extending[10].

The basis for this simplicity derives from the decision, which deliberately breaks with a linguistic tradition referring back to the Arabic linguistic tradition, to treat Arabic words (in the rather naive, but computationally efficient sense of "any sequence of characters separated by spaces") as elements composed in three subparts: a prefix, a stem, and a suffix, the stem being the only necessary sub-element, while admitting a series of zero-prefixes and zero-suffixes.

This simple account is straightforwardly implemented in the (possibly) simplest way, by feeding the system with three lookup lists of, respectively, (a) prefixes, (b) stems and (c) suffixes, together with three compatibility tables between, respectively, (d) prefixed and stems, (e) prefixes and suffixes and (f) stems and suffixes. Entries in the lookup lists are made up of four fields: (i) unvocalized and (ii) vocalized forms of the morpheme, (iii) grammatical category and (iv) English gloss; compatibility tables just list couples of compatible morphemes, all other combinations being incompatibles. Supplementary pieces of information, not employed in the analysis proper but potentially useful for glossing the texts (root and lemma for a group of morphemes), are provided in the stem lookup list in the form of pseudo-comments.

The analysis is performed through a brute-force search of every possible decomposition of words into prefixes, stems and suffixes, by looking up for prefixes (0-4 letters), stems (at least one letter), and suffixes (0-6 letters). Only the unvocalized form of words is taken into account (short vowels and other diacritics are stripped before looking up); all theoretically possible prefix-stem-suffix decompositions are looked up in the tables and discarded if anything is missing. As a result, each word of the input text can be labeled as (i) unrecognized, if no

9. For a fuller description of the model, see Boella *et al.*, 2011.

10. Buckwalter's system has been used in many different projects, mostly in its Java implementation; it is, for instance, included as a morphological analysis tool in the Arabic WordNet Project.

possible analysis is found, (ii) unambiguous or (iii) ambiguous if, respectively, one ore more analyses are returned by the lemmatizer.

This model – whose beauty lies in its very simplicity – is a good starting point for a successful morphological analysis, but shows a number of weaknesses, owing to the original design and its intentional limitations: for instance, while the emphasis on the unvocalized form of the word is relatively justified for newspaper texts and other Modern Standard Arabic (MSA) non-literary texts, namely Buckwalter's ideal target, it is far from ideal for other types of texts, since each information about vowels and other diacritic signs reduces ambiguity.

A second weakness in the original model also derives from the pool of texts Buckwalter worked on: again, only morphemes attested in a subset of MSA texts and their combinations are included in lists and the combination tables of the system, which unavoidably let the system reject or analyze wrongly many words attested in other textual types – to limit ourselves to a single example, the *a-* interrogative prefix is not included in AM, probably because it is rarely found in newspaper texts.

A third, related problem in the original implementation of the system lies in the lack of any stylistic or chronological information in the lookup lists. This gives rise to a number of false positives in the analysis of some textual genres, since many strictly contemporary terms – including foreign named entities – are included and looked up even when a Classical text is analyzed.

To overcome these weak points, our research group devised a number of revisions to the original algorithm, which resulted in a model called "Revised AraMorph" (RAM). The first revision lies in taking into account vocalization as an ambiguity-reducing device, instead than discarding it. This is more complex than it might seem – the entire segmentation phase had to be redesigned –, but it brought significant advantages in performance even for mildly vocalized texts, as most Classical texts are.

To tackle the second weak point, namely the unbalanced, MSA-biased coverage of the Arabic lexicon in the original model, a file with additional stems automatically extracted from Salmoné's (1989) Arabic-English dictionary in its electronic format was added to the system. Moreover, an analysis of some of the most frequent types of errors in recognition added a limited, but important, number of additional prefixes and suffixes. Together to true prefixes, as the interrogative *a-* discussed above, the single more important addition was the set of prefixes, suffixes and combinatory rules for verb imperatives, a category entirely missing from the original implementation (again, as a probable consequence of the newspaper bias of the AM model).

To reduce the third problem hinted at above, namely lack of distinctions of genre and style, we automatically removed items that are likely to correspond to

contemporary foreign named entities, especially proper names and place-names, when dealing with Classical texts[11].

4. Conclusion

This short sketch of some of the problems and the solutions involved in Arabic corpus linguistics, and specifically in the process of annotating Arabic texts just scratches the surface of a complex and still partially understood reality. The electronic revolution, as previous cultural revolutions, is bringing about new paradigms that will take decades to evolve and to change the mental habits of users and researchers. However, the benefits that new ways of analyzing old texts can introduce are so great that the efforts needed to devise new strategies for querying and retrieving information are doubtless worth the effort.

11. A simple example will illustrate the kind of spurious ambiguity introduced in Classical texts by the lack of genre and style distinction in the original AM model: the transcription of the English first name "John" is indistinguishable from *jūn*, "inlet, bay", while this ambiguity is unlikely to appear in a non-contemporary text.

References

BOELLA M., ROMANI F. R., AL-RAIES A., SOLIMANDO C. and LANCIONI G., 2011, "The SALAH Project: segmentation and linguistic analysis of ḥadīṯ Arabic texts", in M. V. Salem, K. Shaalan, F. Oroumchian, A. Shakery and H. Khelalfa eds, *Proceedings of the Seventh Asia Information Retrieval Societies Conference*, Heidelberg, Springer.

BUCKWALTER T., 2002, *Buckwalter Arabic Morphological Analyzer Version 1.0.*, Philadelphia, Linguistic Data Consortium.

BUCKWALTER T., PARKINSON D., 2011, *A Frequency Dictionary of Arabic: Core Vocabulary For Learners*, New York, Routledge.

BURNARD L. and BAUMAN S. eds, 2009, *TEI P5: Guidelines for Electronic Text Encoding and Interchange by the TEI Consortium,* 1.5.0., Oxford, Providence, Charlottesville, Nancy.

GACEK A., 2009, *Arabic Manuscripts. A Vademecum for Readers*, Leiden, Brill.

GREFENSTETTE G., TAPANAINEN P., 1994, "What is a word, what is a sentence? Problems of tokenization", *Proceedings of the 3rd Conference on Computational Lexicography and Text Research*, Budapest, p. 79-87.

HAYWOOD J. A., 1965, *Arabic Lexicography*, Leiden, Brill.

KAPLONY A., 2011, *The Arabic Papyrology Database Guidelines*
[http://www.ori.uzh.ch/research/papyrology/papyrologydocu.html], retrieved 25 November 2011.

SALMONÉ A. H., 1889, *An Advanced Learner's Arabic-English Dictionary*, Beirut, Librairie du Liban.