

An Expert System for the Classical Languages: Metrical Analysis Components

Daniele Fusi

Università di Roma „La Sapienza“

<http://www.fusisoft.it>

fusi.daniele@tiscali.it

Abstract

It would be easy to show that at least two benefits can be dramatically relevant in computer analysis of prosodical, metrical and linguistic phenomena of the Classical texts: the availability of massive and detailed data collected with a uniform method, and the requirement of a well-defined theory formulated in a strongly formalized way to collect them. Of course many of the complex problems involved in performing a serious metrical analysis cannot be fully defined from a theoretical standpoint, but when facing computer analysis one must at least try to find satisfactory solutions for them. This is at the same time the most difficult and intriguing aspect of computer-related solutions applied to humanities: every single theoretical aspect must be fully defined and formalized. This paper presents some aspects of a computer expert system created to perform linguistic and metrical analysis of Greek and Latin texts keeping into account such theoretical problems, and also acting as a subsystem for another project offering a transformational model for automatic inflection of Greek and Latin languages through all the reconstructed historical stages. The metrical system can generate truly interactive metrical ‘editions’ which can be queried by scholars for any combination of observed prosodical, syntactical and metrical phenomena, and several outputs for different uses, with special attention to its integration in other projects like digital editions.

1 Introduction

Almost 40 years ago, when the first computer applications to humanities were just beginning to

appear, A.W. Bulloch could already write that “our understanding of the Greek hexameter [...] would certainly be fundamentally reestablished, if not revolutionized, if all known examples were to be analysed on a computer according to their most important characteristics”¹. Such a view is probably too emphasized, anyway it would be easy to show that at least two points still today can be dramatically relevant in the study of a wide range of prosodical, metrical and linguistic phenomena of the Classical texts: the availability of massive and detailed data collected with a uniform method, and the requirement of a well-defined theory formulated in a strongly formalized way to collect them.

As for the first point, at least since the sixteenth century philologists have been providing many studies devoted to a huge number of aspects of Classical versification in very different contexts, from general or monographical studies to manuals, or even occasional journeys in the realm of metrics only to support a specific hypothesis about any other field of Classics. As a result, even today most of the numerical figures collected by such studies are heavily influenced by their origin and purpose: they refer to portions of selected works ranging from a few tens to some thousands lines, collected by scholars spanning more than two centuries, and often lead (and sometimes mislead) by very different theoretical beliefs². Still, even the manuals refer to some of these studies to present what should be the plain description of the phenomena: it’s a matter of fact that even today the various specific stu-

¹ Bulloch 1970.

² A clear example of theoretical assumptions leading to completely unreliable data and conclusions is a well-known study by O’Neill (1942), which yet is still quoted as the source for many numerical figures.

dies produced in the latest centuries are the only source to get at least approximate figures about them. Of course, the scholarly studies of past centuries are the big foundation of any modern work, but we are facing lots of limitations due to the simple fact that humans cannot spend their lives collecting data from thousands and thousands of lines, syllable by syllable. That's more a machine work.

For instance, the fundamental *Metrik* by P. Maas³ reports violations of Hermann's law or spondaic hexameters every 1000 and 50 lines respectively, but in the English edition of the same work⁴ these figures are clearly undersized to "about" every 390 and 18 lines (i.e. from 0.1% and 2% to 0.26% and 5.56%). Nonetheless, many studies and manuals still repeat the first figures, and there is no way to verify them other than simply repeating the analysis. The fundamental problem here is that all such data come from analysis necessarily limited to very small samples (furthermore coming from different and sometimes outdated editions of the sample text) by many different scholars, who rarely attempt to explicitly state all the implicit assumptions in their data collection method. Yet these are the best sources of data, as in too many other cases scholars have limited themselves to verbal approximations which apart from being subjective are simply unusable for any other purposes (nobody could try to base a new hypothesis on indications like "often", "rarely", "about", "generally", etc.). Further, the numbers themselves can be misleading if we don't test their significance with adequate statistical tools (what happened almost always in past studies, while nowadays many recent studies have realized and applied this principle). Lastly, such tools are invaluable for determining the significance of a phenomenon but are completely useless if the data they are applied to have been collected with non-uniform methods (an non-systematic error in the measurement tool would make any statistical processing unreliable).

This leads us to the second point: method. Here things can become even more uncertain, not only because of different theoretical beliefs, but also because almost no theory is without 'obscure' regions which are left to scholars judgment or common-sense: everyone reading a grammar or a metrics manual is familiar with approximations

like "often", "usually", "etc.", "and the like", but unfortunately a machine cannot operate with such vagueness. Of course many of these problems cannot be fully defined from a theoretical point of view, but whoever wants to apply computer analysis must at least try to find a practical solution for them. This is at the same time the most difficult and intriguing character of computer-related solutions applied to humanities: every single theoretical aspect must be fully defined and formalized with no room for uncomplete or vague statements. To make a trivial sample, just think about the detection of word-ends in a line, implying an (at least practical) definition of what can be considered a "word" in metrical and linguistic terms, and the much debated notion of appositive words: from such a definition derive fundamental assumptions for word-ends, metrical laws and inner metre structure. Anyway, an adequate theoretical basis could offer the practical solutions for a computer tool capable of producing a full prosodical and metrical analysis of any Classical verse, which can prove very useful in providing at least the massive and yet fully detailed data for the *observatio* which should be the foundation of any hypothesis, thus contributing to better define some of the most debated questions in the metrical field. I have already tried to show some examples of such results in papers about even well-known and long-studied phenomena like hexameter's laws and their evolution in time from Homer to Nonnus (e.g. the paradigmatic and syntagmatic nature of Hermann's and Lehrs' laws⁵), or about the linguistic implications of peculiar metrical usages in late-Latin poets like Luxorius⁶.

Finally, when analyzing metrics we must never forget we are dealing with very complex organisms where linguistic material is structured according to several combining factors. When we want to study a single phenomenon we should be able to isolate at least the most important bias factors affecting it, and this requires an extremely detailed and huge amount of data, even beyond the scope of our (or others) specific study. Let's make a trivial sample: consider the dactylic hexameter. Here the combinations of spondaic and dactylic feet produce 32 variants. Of course dactylic and spondaic feet are not evenly distributed along the line, but they vary

³ Maas 1923.

⁴ Maas 1972.

⁵ Fusi 2002.

⁶ Fusi 2004.

with preference for dactyls especially towards line end. For instance, in the very short hexametric text I'm using here as a sample (Aratus *Phaenomena*, 1153 lines) the spondaic feet are distributed as in the following chart⁷:

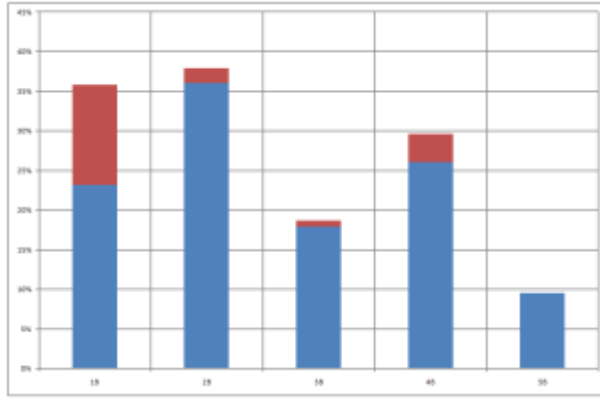


Chart 1: distribution of spondaic feet in positions 1-5

The chart shows the percentages of spondaic feet at each position (1-5 from left to right); the topmost part of each bar with a different color represents the portion of these spondaic feet with a ‘true’ wordend⁸ after them⁹.

Now, consider the distribution of wordends in the most sensitive positions we call *caesurae*: among them the most important is the feminine, which cuts the two short syllables in the third dactylic foot: this of course implies that we must have a dactyl in the third foot. Thus, here we have two combining factors at play: the frequency of wordends at each position in the line, and the distribution of dactylic and spondaic feet. If we want to study wordends in isolation we should remove the bias of the latter on the former, e.g. ‘weight’ the wordends on the feet types.

Chart 2 shows how our results may vary: it represents the frequency of wordends in the hex-

ameters of Aratus from line beginning (=left) to end (=right). As you can see, the peaks correspond to caesurae and the valleys to bridges (e.g. Lehrs and Hermann, to quote the most known). Here the lines represent the absolute percentage of wordends, while the areas represent the percentage calculated no more on the total lines count but on the lines showing either dactylic or spondaic foot at each sensible position. As you can see, things vary noticeably, and the ‘weighted’ percentages are much higher. You can also see in the bottom part of the chart (red in the slides) the distribution of false wordends, i.e. wordends involving appositives, which too were kept isolated from true wordends.

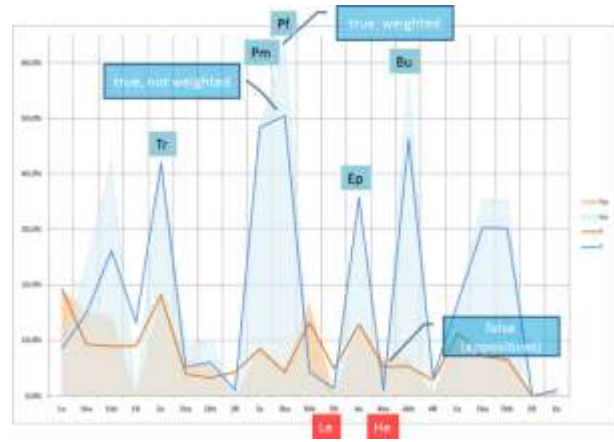


Chart 2: distribution of wordends along the whole line. Labels mark caesurae and some bridges. Areas refer to ‘false’ wordends.

Another very trivial sample can illustrate the interaction among language and metrics: elision has a complex profile involving the types of words it affects, but first of all it implies the obvious condition that there must be a wordend to allow for it. This brings into play the distribution of wordends in the line, on which we should ‘weight’ the distribution of elisions themselves. Let us consider chart 3: it shows the distribution of various types of elided words in the line.

Notice the peaks at the positions corresponding to the caesurae in the third and fourth feet: they look so high that one might be tempted to infer that there is some sort of correlation between wordend in these positions and elision. Yet this is not the case, as we can see if we ‘weight’ our percentages by calculating them no more on the total lines

⁷ This chart as all the data discussed in this paper come from the computer analysis illustrated here, but are limited to a very small sample to avoid more complex discussions on genres, chronology, etc. as my aim here is just so present an overview of this expert system. For the same reason in this paper the charts appear very small as they are meant to just provide a sketch of the phenomena (refer to the accompanying slides for bigger versions).

⁸ For the meaning of this distinction see section 4 (Syntax) below.

⁹ This specific distinction is due to well-known tendencies to avoid a wordend after spondaic foot especially in certain positions, as expressed by hexameter “laws” like Wernicke, Hilberg and Naeke.

count, but rather on the frequency of wordends at each position.

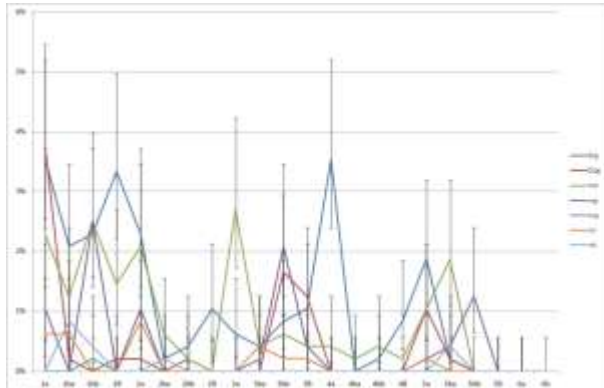


Chart 3: percentages of elisions along the whole line, calculated on the total count of lines examined. Different lines refer to different word classes.

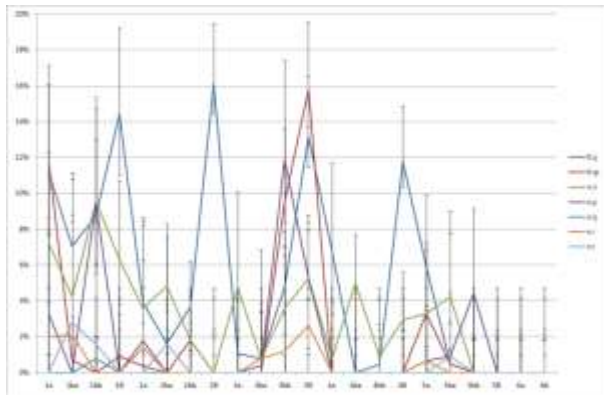


Chart 4: percentages of elisions along the whole line, calculated on the frequency of wordends at each position.

As you can see, things change completely: now at the fourth foot we have a valley instead of a peak. The trivial reason for that is of course that wordend is the necessary condition for elision, and thus wherever wordends are frequent (in *caesura*), we can expect elisions to be frequent, too. Without taking this simple consideration into account we might be fooled by apparently correct data. Of course, this requires us to provide detailed data about elisions and wordends at each single position in the line for each line in our sample: it is precisely here that our machine analysis can prove useful.

These are just trivial samples I chose for their simplicity, but it is easy to understand that in analyzing any aspect of a metrical text we should keep into adequate account all the factors which concur in producing a surface phenomenon. This often re-

quires us to provide much more data than what we could expect at a first glance, even if we just need them to properly view our specific subject of study. In contrast with all the limitations we have summarized here, machine analysis provides the possibility to examine huge amount of texts with extreme levels of detail, collecting data for thousands of lines down to the single phoneme. Even more important, it implies the definition of a highly formalized method, and it grants its rigorously uniform application throughout the whole text sample. This in turn allows us to properly use statistical methods to test the significance of our data and thus confirm or reject the validity of working hypotheses. Finally, a machine-driven analysis can be repeated indefinitely on new texts building on top of existing data, either to update them by using new editions or to add new texts or new types of observations.

Of course, this comes at a price: the machine knows nothing of metrics and language and its analysis is purely formal (and strictly speaking this might not be so bad for prosodies and metrics), so every single methodological detail must be fully defined without ambiguities or approximation. This requires a big effort for both scholars and programmers, but I think it can be rewarding not only for metrics but also in the context of the more general expert system some parts of which I'll try to show in the following overview.

2 System Overview

The metrical analysis system I shortly present here fits into a much bigger picture where highly specialized components (metrical analysis, automatic inflection of Latin and Greek language, lemmatization, etc.) operate side to side with more generic digital editions frameworks. Any account of this system would be outside the limited scope of this short presentation, but this is one of the reasons for the componentized structure and abstraction level of several aspects of this metrical subsystem. For instance, all the components which provide the full phonological and prosodical analysis of a line are completely shared among other subsystems, first of all the morphological one, whose aim is to generate all the inflected forms for a given word in a historical perspective. As any student of language learns since his first grammar, morphology of course requires phonology: know-

ing where syllabic boundaries fall, whether a vowel or a syllable is short or long, where the accent is located, etc. is often a condition for the formulation of several grammar rules. At the same time, metrics shapes most of the same phonological phenomena into less or more complex patterns, and this explains why the phonological analysis components are effectively shared among these different subsystems. Further, many theoretical aspects at their foundation can be generalized at such a level of abstraction that they can be easily applied to both Greek and Latin, or even to any other language. Thus, in this context not only components are shared among subsystems (e.g. phonological analysis serving both metrics and morphology), but also parts of their implementations (e.g. the detection of theoretical syllabic boundaries according to phonematic openings used by any language-specific syllabification function: Greek, Latin and even Italian). Finally, all these components work together in even bigger frameworks, for instance when a digital *corpus* provides sample texts for metrical analysis and this in turn outputs its results in a form which can be easily integrated into the original *corpus*, thus creating a specialized edition built on top of an existing one.

In the following sections I'll present a very short account of the most important components used by the metrical subsystem, leading the reader through the essential stages traversed by my software in its analysis.

3 (a) Prosodies

The first stage for metrical analysis consists in importing the text itself. Typically the texts to analyze are extracted from large digital *corpora* like Packard Humanities Institute (PHI) cd-roms using another software component I created for this purpose: it can read any PHI cd-rom, extract the desired portion of text and fully recode it from *Beta code* into *Unicode* or into any encoding, either standard or not, using a heuristic approach. In the context of the bigger picture illustrated above this component is shared among several other subsystems, like e.g. digital editions, to generate output in any textual encoding and format ((X)HTML, RTF, XAML, XPS, etc.). The same component is also used wherever the metrical subsystems needs to get text input from the user, or conversely to output some formatted text. Again the same component is

used also in the form of a *Word* addin to ease the input of ancient Greek *Unicode* text, thus providing popular word processors like *Word* with the powerful editing and conversion environment specialized for Greek and Latin texts used in my own software (including digital epigraphical editions).

Thus, whatever the input text format and encoding may be (either coming from digital *corpora* or directly typed by user), the software finally gets a plain *Unicode* text to analyze. Of course, this text is almost always somewhat complicated by additional data like e.g. line numbers, layout features (line indent and spacing), special characters (e.g. the *diplai* in the TLG cd-rom text of Homer), etc. The first stage of the analysis proper thus consists in a smart parsing of the input text as we need to remove all the irrelevant 'noise' characters: some (like line numbers) are used to collect metadata, others (like non-textual characters as *diplai*, parentheses, etc.) are just discarded, but in both cases they will be restored at their place when generating an output for the end user. The text filtered in this way is then normalized (for e.g. extra spaces, letters casing, glyph variants, etc.) so that the phonemic analysis can get an input where all the confusing or irrelevant variants have been removed.

A process we may call "phonemization" then occurs after parsing: at this stage, the program uses a set of external phonological parameters (each specific for a given language) to deduce a sequence of phonemes (or allophones) from the input sequence of graphemes. All the required contextual analysis (like e.g. the detection of diphthongs) happens at this stage too, as of course deducing sounds from letters is never a one-to-one mapping (cf. e.g. Latin «x» = /ks/ and «qu» = /k^w/), but often complex algorithms are involved.

All the language-specific data at this stage are kept in a set of XML files separate from the program itself, which tell it how to interpret each letter phonetically (e.g. providing point and mode of articulation, highness, phonological traits like voicing, rounding, length, etc.). This allows the same program to work for different languages like Greek and Latin¹⁰.

¹⁰ At the time of writing the metrical components are ready only for Greek, but the phonological analysis of Latin is the basis for automatic inflection of this language.

The program uses this information to build a structure which will be the basis for every successive analysis: the text is represented as a chain of segments, each linked to any number of data stored in different ‘layers’: such layers tell whether a given segment is long or short, where syllabic boundaries fall, how words are connected, how syllable weights relate to metrical schemes, etc.: all the data calculated by the system in the next stages are stored in these layers, which can grow indefinitely so that we are free to link specific information to each phonological segment without complicating the analysis with the insertion of non-textual material among them.

As for the analysis of Greek graphemes (and even more for Latin, where all the letters are *dichronae*), further difficulty arises from the so-called *dichronae* letters¹¹ Α Ι Υ, which can represent either short or long vowels. The software is able to tolerate such lack of information, but it strives to find out every possible bit of data: it uses word accentuation to infer lengths, but it also recurs to a sort of ‘prosodical lexicon’ which can be queried during analysis. This lexicon is generated and maintained by the software itself using contextual analysis (see below under section 5).

A second issue comes from a great deal of prosodical variants which can severely affect the verse: for instance, the alternative syllabic divisions of some consonant groups (e.g. the so-called *muta cum liquida* group and more rarely other sequences), the potential redoubling of some consonants in specific word positions (e.g. initial nasal or liquid), and finally hiatus, which may shorten a long vowel or diphthong (*correptio epica*). All these parameters may optionally affect the line, and again are stored in separate XML files, specific for each language (or for each variant of a language, which is the case for the different Greek literary dialects used in various genres). Such files tell the software that some consonantic sequences or single consonants, eventually in specific word positions, might or not trigger a different syllabification by virtue of tautosyllabic measurement or reduplication, with different levels of probability. At this stage the software takes no specific action in such cases and usually applies the standard treatment, but it keeps this information in a layer

¹¹ For this definition cf. Rossi 1963.

linked to the related segments, so that the metrical component will be able to use it later.

Finally, the prosodical subsystem detects syllabic boundaries by analyzing the opening of each phoneme in the segmental sequence defined earlier. To do this it relies on Saussure classical model of the so-called phonological syllable, which is a universal model defining a syllable as the distance between the two segments with the lowest opening value. As the program has already collected data about each phoneme, including its opening, it can easily infer the theoretical syllabification, which can be output to the user for diagnostic purposes as shown in chart 4.

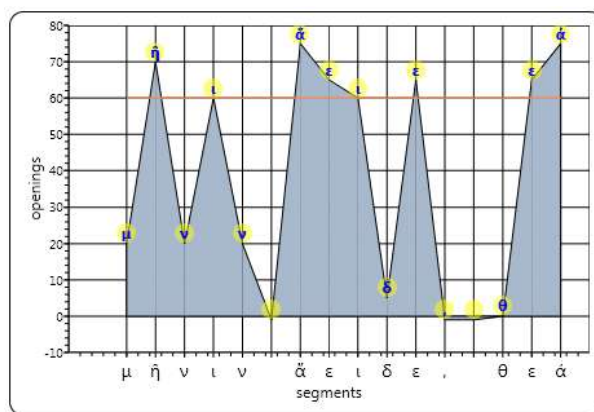


Chart 4: computer analysis of phonematic openings in the first words of *Iliad*.

This chart shows the syllabification of the first words of *Iliad*, 1. As you can see, the openings drive the detection of syllabic boundaries, with their subsequent adjustments for Greek language. Of course, this model is just an approximation and each language requires its own specific adjustments¹², so after a first detection syllabic boundaries are adjusted by specialized functions.

Once syllabic boundaries are in place, it’s relatively easy to detect (wherever possible) syllable weights, as of course any closed syllable is heavy, while in open syllables the weight depends on vowel length.

¹² Think e.g. of a sequence like Latin *stare*: here the first “phonological” syllable would be just *s-*, as it’s followed by a lower opening segment (*t*, a voiceless plosive). Even if this has some support from the evolution of the language itself (cf. the prothetic vowel in Italian *istare*), the “phonological” model here must be corrected with the “phonetic” one, whence the two syllables *sta.re*.

This defines a very efficient model for phonological analysis: the same software components can be shared among several specialized analysis subsystems, e.g. for Greek, Latin or Italian. All the components which parse a text, phonemize it, add special markings for optional prosodical treatments, and provide a generic syllabification are fully shared. What is specific to each single language is relegated to external XML parameter files. Only the last stage, which adjusts syllable boundaries, is necessarily specific for each language.

4 (b) Syntax

After prosodies, the second big subsystem is the syntactic one. It deals with words classification by distinguishing among the so-called “lexical” words and the crucial class of appositives, which typically have higher textual frequency, lower lexical frequency and a very small size. They in turn include words with and without accent (clitics), which finally part into enclitics and proclitics according to their connection to the left or right.

Any serious metrical analysis cannot do without this words classification, as it literally shapes the verse for its “inner” metric (wordends, bridges, etc.: as already Maas pointed out clearly, a sequence of graphical “words” like *kai tòn patêra mou* is just one ‘linguistical’ word). This is not the place for even a short discussion of this complex subject, but you can immediately grasp the big difference between analyses which take into account this problem and those which don’t by just giving a look at charts 5-6.

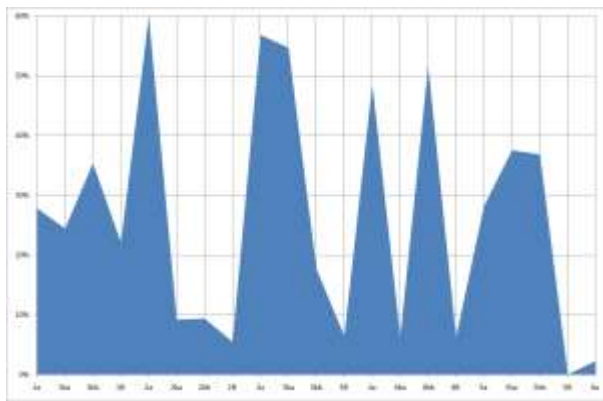


Chart 5: distribution of graphical wordends

Chart 5 shows the distribution of purely graphic (‘false’) wordends in our usual sample text, while Chart 6 shows the (unweighted¹³) distribution of ‘true’ wordends.

As you can see, peaks in Chart 5 roughly correspond to the well-known caesural places, but they shape a picture where secondary caesurae like the trithemimeres would appear higher than primary caesurae like the penthemimeres (cf. the left-most peak), and even the most severe bridges (like Hermann or Lehrs) would allow for an unexpectedly high percentage of violations (cf. the valleys at about the middle of the horizontal axis). Of course this picture is completely distorted by the superficial treatment of each graphical space as a true wordend boundary, as if in English we were considering as full ‘words’ the articles “the” and “a” or the preposition “in” in a sentence.

If instead we apply a more realistic linguistical analysis to our sample text, by taking into adequate account word types and their *liaison* direction and accentuation, what we get is Chart 6: here you can easily see that the curve visually defines the expected shape for the (Hellenistic) hexameter: for instance, peaks to the left and right have been lowered and the prominence of the penthemimeres caesura clearly emerges; also, the feminine caesura looks more frequent than the masculine (cf. the top-right angle in the central peak against the top-left one in Chart 5); and finally, bridge positions show clearly deeper valleys corresponding to a much lower percentage of violations. This time the chart truly provides a visual representation of the inner structure of the metre with its well-balanced distribution of wordends peaks.

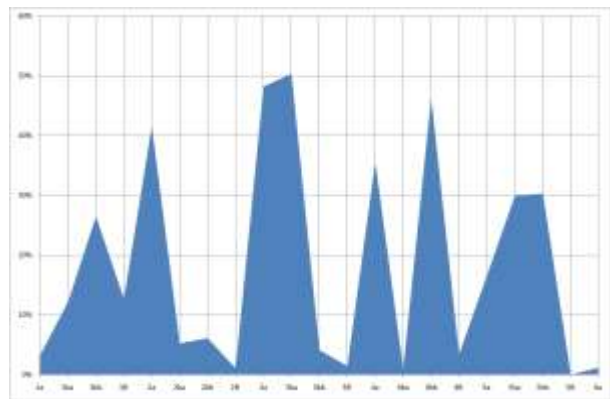


Chart 6: distribution of ‘true’ wordends (same sample and scale of Chart 5)

¹³ For this concept see the above discussion about Chart 2.

This trivial sample should be enough to show the crucial relevance of a linguistically serious treatment of wordends. Of course, an adequate and at least pragmatically usable definition of appositives requires a lot of effort and the combination of phonological, syntactical, semantic, lexical and rhythmical factors: but the complexity of such a definition should not be an excuse for avoiding it at all and simply sticking with graphical words, as the same common sense which does not trouble the native speakers of a language in defining a “word” shows us that we could never place at the same level “words” like *te* or *perí* and “words” like *polýtropos*. Of course, it’s not just a matter of mechanical classifications: it would be wrong to rely on a single aspect like monosyllabic or bisyllabic body, presence or absence of accent, lexical character etc. The status of a word comes from the combination of several factors, and we must also face with a heterogeneous graphical system (which of course is the only source of data for the machine) which often hides further complexities (e.g. think of the purely graphical accentuation of proclitics).

The analysis is further complicated by the fact that words must be analyzed in their context, which may severely alter their surface shape: think for instance of sequences of clitics with eventual development of enclisis accents, or of the phenomenon of barytonesis, and of even more complex syntagmatic phenomena like the so-called “continuatives”. To make a trivial sample, just think of this three “words” sequence: *perì d’ ouranón*. From left to right we have a proclitic word (no accent and connection to the right; the accent we note here is merely graphical, and the software must know it), followed by an elided enclitic (which too has no accent, even if it bears a graphic one in its unelided form) and finally by an orthotonic lexical word. Now, in this sequence the short body of the elided *d’* isn’t enough to stop the preceding preposition *perì* to connect not only with this particle, but also to *ouranón*: in this sense we say that *d’* is a continuative enclitic, as it allows the connection to the right of the preceding preposition to continue even after itself to make a bigger group¹⁴.

Thus, the syntactic subsystem uses very complex syntagmatic algorithms to take into account all the surface changes of these words and detect

their nature. During analysis, it takes all the data about appositives (paradigmatic form, ‘true’ and graphical accentuation, direction of connection, continuative potential, etc.) from a relational database, and it enables the metrical subsystem to deal with some 16 types of wordends, as defined by the combination of four factors: true or false – i.e. merely graphical – wordend; presence of hiatus between words; presence of aspiration in hiatus; presence of elision in hiatus. All this information¹⁵ is stored as usual in the data layers linked to the text segments.

5 (c) Metrics

The third subsystem after prosodies and syntax is finally metrics. Until now, the system has taken a text, parsed and converted it into *Unicode*, defined its phonological values and syllabification, and classified its words: all this has been done by accessing several parameters from XML files and relational databases.

The metrical subsystem too takes its parameters from an external XML file: this time it contains the definition of any verse design we want to use (either stichic or strophic: hexameters, elegiacs, trimeters, etc.). Its job is to generate all the possible implementations of each verse design, and match the sequence of syllabic weights coming from earlier analysis to them. This is not a straightforward process, as apart from potential ambiguities (wherever a sequence of unknown weight syllables is long enough to cause them), the matching process itself is designed so to trigger relevant changes in prosodies, which in turn imply a new scansion. For instance, as required by context it could move syllabic boundaries in a group of *muta cum liquida*, or redouble a consonant, or shorten a long vowel in hiatus, etc. In some cases, because of the potential lack of enough data, the program can prompt the user to resolve ambiguities whenever more than one verse design implementation might theoretically fit the given syllabic sequence.

¹⁵ This subsystem also copes with some language-specific issues like e.g. the contextual disambiguation between **to-* derived forms in Greek: in this language, forms derived from IE **to-* may be anaphoric pronouns (orthotonic), articles (proclitic) or relative pronouns (prepositive). Six rules are provided to achieve a semiautomatic distinction of anaphoric and relative / article values. For these rules I draw data mainly from P. Monteil 1963.

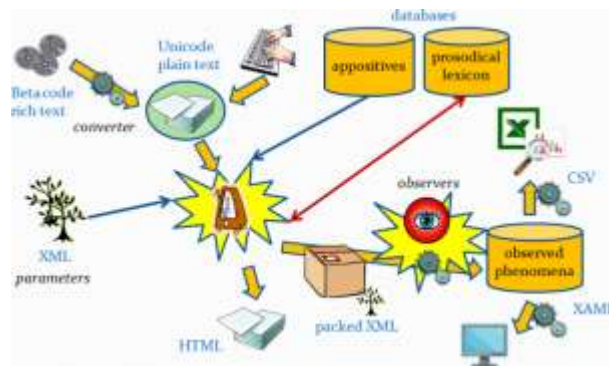
¹⁴ For this concept see especially Cantilena 1995.

At this stage the software also makes its vowel lengths deductions according to the metrical context, thus feeding the prosodical dictionary cited above. This dictionary is stored in a relational database, which (whenever the user activates this option) is continually enriched by the analysis process itself. Initially empty, the database is filled by the software whenever its metrical scansions give any clue for lengths deductions. After any deduction the corresponding word with its deduced lengths are stored in this dictionary, so that the next time the program finds the same word it will be able to use the deduced lengths and thus reduce the lack of information. This allows the program to ‘learn’ from experience: the more lines it scans, the more words are added to the dictionary¹⁶.

6 Output, Observation and Evaluation

This three-tiers system generates several outputs: some are designed to be immediately displayed to the user (e.g. for diagnostic purposes) and are based onto XAML, an XML dialect; other produce richly formatted and detailed (X)HTML files for reporting; and finally all the data coming from analysis (from prosodies up to metrical scan-

sion) are stored in a standard packed-XML format¹⁷ ready to be read by the next subsystem.



The report files generated by the metrical analysis subsystem are mainly used for diagnostic purposes, but they can also constitute the foundation of a full ‘metrical edition’ of any given text: just think of a poetical text where each line has the full detail of prosodical, syntactical and metrical analysis summarized in a synthetic table and available to user interactivity. For instance, in XHTML or XAML format this output contains a full text where the user can read each line with its syllable weights and boundaries and metrical scansion; also, by just hovering the mouse on the text he can get details about each single syllable.

Anyway, the most important output is represented at this stage by the packed XML data which include all the machine-readable information produced by the previous analysis. This information is the input for the fourth big subsystem, which reads the data packed in XML and literally observes them to detect any kind of phenomena we may be interested into, at any level: prosodies (e.g. *muta cum liquida*, redoubled consonants, accents distribution, etc.), syntax (word types distribution and connections, etc.), metrics (verse instances, “laws”, *caesurae*, bridges, etc.). This architecture is modular, and new specialized ‘observers’ can be added at any time. This allows us to reuse the same analysis data for observing new kind of phenomena, as it often happens that the study of data makes it necessary to observe new things, or to view existing data under another perspective (cf. our discussion of the bias factors affecting the phenomena under study). At this time, I have implemented 23 ob-

¹⁶ Of course, the software is able to generate the paradigmatic form of a word by removing all the phosyntactic or even graphic modifications that may affect it in a given context (e.g. barytonesis, enclisis accents, letter casing, etc.). It may also happen that a word with more than one unknown length vowel gets only some of them specified by metrical context. The software can handle such cases and also retrieve later the same word to further specify its still unknown lengths as soon as they eventually get specified by other contexts. Further, we must also take into account the possibility of more measurements for the same word form (e.g. *kalós* with either short – like in Attic – or long *a* – like in Ionic –, as a consequence of the different syllabification of the original group *-lw-* and the subsequent compensatory lengthening), or of homographs (e.g. *eruthrá* which could be either feminine singular – long *a* – or neuter plural – short *a*). As for several other cases, fully describing the analysis process would be outside the scope of this paper, but here it’s enough to remark that such problems have been taken into proper account so that we can rely on the most accurate analysis possible. Obviously even the most accurate machine analysis might not be perfect, but the availability of huge amount of data coming from thousands of lines analyzed with a rigorous and uniform method together with their proper statistic evaluation should allow us to regard such objections as generally irrelevant.

¹⁷ The choice of packed XML is first of all dictated by the desire of reducing the output size, but it also offers a powerful standard to store more structured XML data files.

servers which collect some 200 data for each single line. This means that for a *corpus* like the one I analyzed using a previous generation of this system (about 90,000 lines) we could get more than 1 million data.

All these data in their full detail are stored by such observers in another relational database. Once this database has been filled with observations, another software component is used to allow users to query it in whatever form they prefer. Users can ask the program to provide the details (with text line by line) for each combination of any phenomena, or generate synthetic reports by aggregating and filtering data as requested. The program offers a user interface where users can visually build their query by combining any of the phenomena observed with logical operators (AND, OR, NOT), get a subset of them or group them in any desired way. This allows a tremendous flexibility as users have at their disposal a truly digital metrical ‘edition’ of texts which they can interactively query at each time for any given data they are interested in. Instead of a model where data (whatever their detail may be) are output once for all uses, this system implements a model where they are selectively published at each single user request. This of course is the only way of making such a huge amount of detailed data usable to end-users, who may want to use them for different purposes each time they are studying a specific phenomenon or combination of phenomena. This also allows us to fully appreciate the bias that several factors may exert on surface phenomena, as sampled at the beginning of this paper.

The program outputs here range from formatted (X)HTML with fully highlighted text to detailed data reports in standard formats like XML or CSV. Finally, these files are imported by third-party software specialized for statistical analysis and charting, for instance spreadsheet applications like *Excel*. Here we can test data for their significance, and emit hypotheses about the explanation of phenomena. Armed with this knowledge we can return any time to the previous program and ask for new data which can further enlighten obscure points and newly arising questions. The system thus grants a fully interactive analysis process, a true laboratory for metrical and linguistic analysis.

7 Technical Overview

Any technical detail would be outside the scope of this paper, but given the context of this congress it will be useful to sum up the main aspects of the implementation of the system illustrated above.

As a programmer and philologist I have personally conceived and implemented the whole system, which as shown above has several other applications, ranging from full-featured truly digital editions with any sort of specialized content to many more commercial-oriented applications (multiple language dictionaries, literary *corpora*, search engines, thematic dictionaries, complex redactional applications, etc.).

The software is fully written in C# in the context of *Microsoft DotNet Framework*. Its most recent portions and all the components which would get essential benefits from the migration have been upgraded to the latest available version of the framework (3.5, which comes especially handy when dealing with XML data using LINQ and with complex user interfaces using WPF). All the components which build up the system are implemented as several separate modules (assemblies hosted in DLL’s), so that the engine is completely independent from the user interface.

User interfaces are almost completely built with WPF, apart from older components still using *WinForms*. External data are stored in XML files or in relational databases implemented with *SQL Server* and accessed via ADO.NET or LINQ. Text encoding is always *Unicode*, but the output can be generated with any even non-standard encoding.

Finally, as pointed above the output is variously represented by SQL Server databases, XHTML + CSS, XAML, XPS, XML. XML transformations are done via XSLT and eventually C# extensions.

References

(The references are strictly limited to the works expressly quoted in this paper)

- Bulloch A.W., *A Callimachean Refinement to the Greek Hexameter*, «CQ» n.s. 20 (1970) 258-268.
- Cantilena M., *Il ponte di Nicanore*, in Fantuzzi M., Pretagostini R. (cur.), *Struttura e storia dell’esametro greco*, 9-67, Roma 1995.
- Fusi D., *Appunti sulla prosodia del Lussorio di Shackleton-Bailey: alcune questioni di metodo*, in Bertini F. (cur.), *Luxoriana*, Genova 2002, 193-313.

- Fusi D., *Fra metrica e linguistica: per la contestualizzazione di alcune leggi esametriche*, in Di Lorenzo E. (cur.), *L'esametro greco e latino: analisi, problemi e prospettive*, Atti del convegno di Fisciano 28-29 maggio 2002, 33-63, Napoli 2004.
- Fusi D., *Edizione epigrafica digitale di testi greci e latini: dal testo marcato alla banca dati*, in Ciula A., Stella F. (cur.), *Digital Philology and Medieval Texts*, Pisa 2007, 121-163.
- Maas P., *Griechische Metrik*, Leipzig/Berlin 1923; id., *Greek Metre*, translated by Hugh Lloyd-Jones, Oxford 1972.
- Monteil P., *La phrase relative en grec ancien. Sa formation, son développement, sa structure, des origines à la fin du V^e siècle A.C.*, Paris 1963.
- O'Neill E.G., *The localization of metrical word types in the Greek hexameter*, «YCIS» 8 (1942) 105-178.
- Rossi L.E., *Anceps: vocale, sillaba, elemento*, «RFIC» 91 (1963) 52-71.