Computers and Text Editing: A Review of Tools, an Introduction to UNITE and Some Observations Concerning its Application to Old Spanish Texts
FRANCISCO MARCOS MARÍN

Some basic issues of textual criticism. Texts derived from lost originals through indeterminate numbers of copies present the textual critic with the problem of reconstruction. The form of a text should be thought of as a pattern. The editor rebuilds that pattern according to the total amount of data available to him. He in fact possesses much more historical information than the author, who, in the moment of creation may lack an overall understanding of the work he is composing. In the rebuilding process, numerous formal, historical, aesthetic, authorial, and sociological considerations guide the editor’s decisions to a greater or lesser extent. While he may subscribe to Thorpe’s maxim (1970:30): “the ideal of textual criticism is to present the text as it was intended,” he must reckon with the fact that his raw materials involve copies whose texts are the result of a sometimes considerable number of prior editors or quasi-editors (i.e., scribes and composers), whose formal criteria for the reproduction of texts were anything but uniform.

Computerized editing avoids the sort of subjective preferences that have often led editors astray. This is not to say that human intervention does not remain a constant necessity; the editor must either mark the relevant textual witnesses in advance by tagging significant features for recognition by the machine, or answer programmed queries about the text which the machine presents to the operator in some sort of menu. The most highly developed

1. I am indebted to Joseph Feustle, Roger Wright, Charles Faulhaber, Arthur Askins, and Jerry H. Craggock for their help in polishing the English version of this article. I am also grateful to Rodrigo Reyes Lirita for his help in the preparation of the first draft in English, and to Juan de Dios Godot for writing the main programs of UNITE. UNITE is a registered trademark of Francisco Marcos Marín, Madrid, Spain. The system was developed thanks to fellowships granted by IBM and the Alexander-von-Humboldt Stiftung; further elaboration was made possible by a grant from the Comunidad Autónoma de Madrid and the hospitality of the University of California, Berkeley, during the summer of 1989.
program for text pre-processing through tagging is TUSTEP (see below, 000), while the most advanced menu-driven program is UNITE, the chief

topic of this paper. In both cases the procedures to be followed are spelled
t out explicitly and the computer follows them without deviation.

Explicit procedures followed without deviation do not lead the editor
to the unattainable ideal of the “definitive” edition. Like Shillingsburg
(1986: 170), I favor banishing the word “definitive” from the vocabulary
of text editing. However, I submit that by following a precise set of cri-
teria, editors can achieve a better text than by any other method. This preci-
sion is rendered vastly more explicit by the computer, but it cannot be re-
teated too often that the computer is unable to do anything on its own
initiative. It is the editor who makes the decisions and who chooses the path
to be followed.

Transmission. Texts were copied from models that included corrections
and glosses meant to complete the text but which rendered it progressively
more complex. At the same time, copies became more and more frequent in
order to satisfy increasing demand. Copyists generally followed a written
text. Copying from dictation was not customary, although there exist some
reliable accounts of this practice, which was resorted to when several repro-
ductions of a masterpiece were urgently needed, with little concern for quality
as opposed to quantity. The copyist copied his model in toto: text, glosses,
and any marginal notes, including at times even those scribbled by a previous
owner without any connection to the text.

One often finds in the literature to “originals.” Generally
speaking, originals did not exist per se. Authors prepared the final state of
their work on slips of paper or parchment, or on pieces of inscribable ma-
terials, such as wax and slate. Organized into a single composition, they were
the basis for the first clean copy, perhaps an autograph, in the author’s own
handwriting. The first handcopy of an autograph, the apograph, usually served
as a source for successive reproductions, often divided into pieces, pecio.
From the beginning, errors multiplied rapidly. Some were simple misread-
ings; others were conscious emendations of varying quality made by copyists
convincing of their superior knowledge. Along the chain of transmission,
copyists normally reproduced the mistakes and corrections in their mod-
els, a circumstance that philologists exploit in order to arrive at plausible
reconstructions.

Textual criticism. Modern editors not only preserve, restore, explain,
and transmit texts; they also establish the genealogy of the MSS in the form
of a tree or stemma, positing hypothetical MSS where needed in order to
give a coherent account of the relationships among the extant MSS. But
the editor does not follow the path in the same direction as the text, from
the original or apograph downwards. Rather, he starts with a text that at-
tracts his attention, and moves “up,” pursuing the impossible goal of arriving
at what might have been the “original.” Until the 19th century, this process
was largely intuitive. Carl Lachmann (1793–1851) established a methodology with two comprehensive steps, recension and emendatio. The MSS are gathered and collated (collatio) in the recension, then, by comparing them and selecting the best readings at each juncture, the text is traced back to the archetype. There may be more than one archetype, depending on the complexity of the transmission. After the editor has gone as far as the textual variants can take him, any further adjustments to the text come under the heading of emendatio.

The archetype may be taken for the original whenever the latter was corrected and revised by the author, but this is not usual. Lacking any sound reason to pursue the reconstruction of a hypothetical original, the editor must resign himself to the reconstruction of an archetype. In fact, he usually cannot go beyond the "common predecessor nearest to the tradition," also called the subarchetype. The flesh-and-bone editor cannot consciously mechanize every process that different copyists carried out unconsciously. Consequently, when the number of copies is overwhelming, it becomes physically impossible to deal with the entire complexus of variants in any systematic fashion. Hence the need for mechanical devices that can be programmed to search and sort through masses of textual data which do not require individual, judgemental, that is, necessarily human decisions.

Some steps back in recent history. The application of computers to collation and recension is not new, and there is considerable methodological variation (Hockey 1980, chap. 7). The era of computational textual criticism began in the sixties, when several experiments in collation were undertaken with the aid of programs written in FORTRAN or even in machine language. They essentially compared two texts line by line and thus may be considered the predecessors of system commands such as CMS "COMPARE" or UNIX "DIFF." Second generation systems could collate prose texts: OCCULT, the Ordered Computer Collation of Unprepared Literary Texts, written in SNOBOL4, was available at the beginning of the seventies (Petty & Gibson 1970). The collations had to be checked by hand, and the percentage of errors (approx. 10%) was excessively high. Working independently, Cabaniss (1970) and Gilbert (1973) both wrote programs in PL/1 to deal with medieval MSS.

The year 1972 marks the beginning of the "third generation," which took advantage of recently developed text editors and word processors. Shillingsburg (1986) cites an Apple IIe collation program by Vinton Dearing (UCLA), built on previous mainframe work, microcomputer programs by Gary Stringer (University of Southern Mississippi), a TeX-related program by Thomas Faulkner (Washington State University), and, with regard to computer-assisted typesetting, CACTUS and CINDEX by David Chestnut.

1. A copy of the program and documentation can be obtained from Stephen D. Dumont, Pontifical Institute of Mediaeval Studies, 59 Queen's Park Crescent East, Toronto MSS 2C4, Canada (e-mail address: DUMONT@VM,EPAS,UTORONTO.CA); NORM works only on texts produced with WordPerfect 6.2. The description given in the text was adapted from that prepared by Dumont for HUMANIST, the electronic bulletin board for computing and the humanities.

2. CMS is the Control Monitoring System of the IBM VM/CMS (mainframe computer); UNIX is an operating system developed at the AT&T Bell Laboratories.
ant, allowing the user either to correct an error (in the keyboard mode) or to write the variant to a file, which becomes the apparatus file. The program runs well and is fast enough to be useful. Semi-automatic text collation is effective, and the program is easy to use. Its main drawback is its restricted capacity: only two files can be collated simultaneously.

The typical format of the apparatus is as follows: 5

```
URICA: User Response Interactive Collation Assistant

TEXT 1 : C:GRIMM.TX1
TEXT 2 : C:GRIMM.TX2

INSERTION
P001L01W08 his <<< wife
P001L01W08 his <<< lovely >> wife

TYPOGRAPHICAL ERROR REPLACEMENT
P001L02W21 which <<< over-looked >> a
P001L02W22 which <<< overlooked >> a

DELETION
P001L03W07 of <<< lovely >> flowers
P001L03W06 of <<< flowers

REPLACEMENT
P001L04W01 and <<< nobody >> dared
P001L04W01 and <<< no one >> dared

INSERTION
P001L04W11 a <<< very >> powerful
P001L04W12 a <<< very >> powerful

REPLACEMENT
P001L04W17 by <<< everybody >> One
P001L04W19 by <<< everyone >> One

DELETION
P001L06W13 eat <<< some of >> it.
P001L06W13 eat <<< >> it.

The case for CASE. During his long involvement in The Thackery Project, which is preparing a critical edition of Thackery's works, Shillingsburg has developed a system that achieves a limited kind of collation of prose texts. The CASE (Computer Assisted Scholarly Editing) program is designed to assist in all aspects of the production of critical editions, from text comparison to the preparation of textual apparatuses and typesetting. Originally the programs were implemented for the UNIVAC 1100/series and for IBM mainframes. They have recently been converted from PL/1 to Pascal in order to make them available as a series of menu-driven programs for the IBM PC and compatibles. I have also heard that a UNIX version is available.

According to Shillingsburg (1986: 135), there are four basic initial requirements for text editing: (1) Each transcribed text is input once only. No output, either of text or apparatus, is ever required. (2) The system must work from the beginning of the collation to the end of the typesetting process. (3) The scholar must exercise the utmost ingenuity to get the most out of the computer, avoiding "silly repetitive work." The lesser the mechanical part of the scholarly job, the better. (4) "Every stage in the process must be interruptible, reviewable, revisable and, if necessary, re-doable."

The first step is to prepare the texts according to the program's specifications (CASE 1983). When two texts are ready, collation may begin. RUN-COLLATE prepares a series of startup programs that run successively unless they find an unsolvable problem. PAGINATE labels copies with pages and lines numbers; PRESCAN and MATCH assure the matching of texts at a series of predetermined starting points. COLLATE then carries out the collation and creates a file of variants. Each variant receives a label identifying its source text, page, and line number for easy retrieval in subsequent steps. CASE cannot directly solve major gaps, such as missing pages or transposed paragraphs. Confronted with a major textual discontinuity, it skips to the next starting point and traces backwards until it gets lost again. While not a definitive solution to this problem, it is an ingenious one. PRINT-VAR prints the variants in a pre-specified format. If MSS with diplomatic and paleographic idiosyncrasies are to be collated, it is possible to use programs which take account of insertions, emendations, etc. Packages built for other purposes, such as OCP, the Oxford Concordance Programme, may also be used at this stage.

The second stage of CASE is the handling of the critical apparatus. Because the collation was done by pairs, the CONFLATION program is required to merge the variants from different lists into a single one. The operator may, of course, delete unwanted entries. Superfluous blank space is deleted by STRIP-VARS. Proofreading against source texts and the addition of notes and remarks by the editor may be done at this point with the help of a word processor. Finally, the entire edition—text, notes, and apparatus—can be photo-typeset using the existing computer files.
TUSTEP. The "Tübingen System of Text-Processing Programs" is composed of routines for editing prose text and, in general, for processing scholarly information of many kinds. All programs work in the batch mode. The operation of each program can be independently controlled, and the programs may be diversely combined to solve a wide variety of problems in handling scholarly texts, including those in non-Roman alphabets. Suitable tasks for TUSTEP include editing and analyzing historical sources, automatically collating different versions of a text, as well as preparing and publishing indexes, concordances, bibliographies, and critical editions. A large number of editions have already been prepared with this package; its usefulness is therefore beyond question. In 1985 it was recognized as "Forschungsschwerpunkt 08 des Landes Baden-Württemberg," which meant that substantial official support was forthcoming for the following five years, support which made possible, among other things, the extension of the routines to many different operating systems.

The preparation of the edition begins with visual and automatic collation, spell-checking, and correction of typing errors. The editor uses up to 29 "tags," i.e., diacritical marks that guide the computer in the tasks of collating, preparing footnotes and building the critical apparatus, or in maintaining the line numbering of different kinds of editions. Two of the tags are general, six belong to the "base text," i.e., the text serving as a basis for comparison, while 21 are used in "manuscript texts," i.e., in the different texts to be compared. No extensive computer knowledge on the part of the editor is required. Except for the basic operating words (LOGON, LOGOFF) and the instructions to load and run TUSTEP, everything is provided by the system.

7. The most recent information I have seen is contained in a draft dated 15 February 1988, which I have summarized from Lancaster & McCaig 1988; for more details see Gellert 1980 and Flett 1980. The system was developed by Wilhelm Ort and Volker Schüle, Zentrums für Datenverarbeitung, Abteilung Literatur und Dokumentation, Univeristät Tübingen, Brunnenstrasse 27, D-7400 Tübingen, Federal Republic of Germany (e-mail address: ZRSZOT@DTUDZU.2.BITNET). TUSTEP runs on OS/1100 (Sparc UNIVAC), MVS (IBM), VM/CMS (IBM), and RS3000 (SILMENS). A (reduced) DOS version for PCs is now available.

After this article went to press, I received word that the Computers and Manuscripts Project at Oxford has finished the first version of COLATE, a text-comparison program designed for the Macintosh environment. According to Peter Robinson (Humanist Discussion Group, vol. 4, no. 1248, 17 April 1991), COLATE can work simultaneously on up to 100 textual witnesses in either prose or verse and of unlimited length (the only requirement is that each witness be divided into blocks no longer than 32,768 words each), deal with heavily marked-up texts, and provides facilities that allow the scholar or editor to tailor the collation and its output as needed. The program functions interactively, with the collation appearing in a window as the editor watches. The editor may intervene at any point to alter the collation by regularizing individual words or phrases according to a broad variety of options. Once finished, the collation can be output as a text with up to five levels of aggregation in a variety of formats, including electronic hypertext. COLATE is available from the Computers and Manuscripts Project, Oxford University Computing Service, 13 Banbury Road, Oxford OX2 6NN UK. For more information contact Peter Robinson at the same address (e-mail: PETER@VAX.ox.ac.uk).

TUSTEP's major functions:
1. EDIEREN assists in the editing of the text, including batch mode correction for cases in which interactive correction is not efficient.
2. VERGLEICHEN compares different texts, collating them and recording variants.
3. BEARBEITEN accomplishes the manipulation of parts of a text, including calculations based on numerical values (e.g., calendar dates) already in the text or derivable from it.
4. REGISTER-VORBEREITEN generates and maintains cross-references and prepares index entries.
5. SORTIER-VORBEREITEN sorts elements into lists and defines various criteria such as collation sequence and alphabet.
6. REGISTER-AUFBEREITEN is responsible for the combination of logically continuous sections of text and the adequate placement of references in the combined text. It also records word frequencies.
7. DRUCK-AUFBEREITEN and LICHTSATZ-AUFBEREITEN prepare output for any one of several different devices, including laser printers, microfilm cameras, and photocompositors with automatic numbering of lines, pages and cross-references.

UNITÉ, or the search for an expert system. UNITÉ consists of a set of programs and commands which help the user to prepare a critical edition by automatically collating different versions, i.e., MSS, of the same work (limited to poetic texts, in its present state). The collation puts the stanzas in order, neutralizes graphemnic variants, regularizes word order and word division, and differentiates and combines function and content words. The program suite attempts to provide a method for reconstructing the portions of the text that the MSS have in common and for choosing between different possibilities when the MSS do not agree. UNITÉ has proved useful with Old English MSS, but it is not designed specifically for Spanish or for ancient or medieval texts. UNITÉ compares different versions of a verse text line by line in order to obtain a unified version; it is not envisioned as a final solution to the problems of textual editing (the length of texts, varying spellings, the existence of different versions, etc.), but rather as a tool that releases the editor from mechanical tasks, allowing him to concentrate on those phases of the process where his expertise is essential. The larger the computerized part of the job, the better, but the editor always remains in control of the process, establishing acceptable editorial criteria and overriding when necessary readings transmitted by the majority of the textual witnesses.

The main characteristic of UNITÉ is that the editor need add no tags or labels to the text being transcribed; in fact, no pre-editing of any kind is required. For Spanish MSS, I recommend the transcription system of the HSMS (cf. Mackenzie 1986), which, by representing medieval and other non-standard graphemes with digraphs, achieves a reasonably close representation of originals using only the lower ASCII character set, a virtually universal standard among different computer systems. UNITÉ, however, can cope with any system of transcription. The current package is designed for
the simultaneous collation of up to six versions of a text, but can be enlarged to operate on more than six. The unit of collation is the stanza, understood as a set of lines of text set off by blank lines. This type of unit permits the scholar to detect and solve problems of misplaced lines inside a stanza. It is not necessary for the different versions to have the same number of stanzas, nor for them to be arranged in numerical order, since various utilities can format the texts and put them in the right order. Nor is it necessary for the stanzas to have the same number of lines, provided that this number \( n \) is between 1 and 5 (\( 0 < n < 6 \)). The only format specifications required of the input texts are the following:

Text identifiers placed at the beginning of the files must contain the same number of lines before the first stanza in all the versions to be collated. These lines contain the name of the manuscript or whatever other indications the scholar may choose to place there. The content as well as the number of these lines is irrelevant for the program, provided only that the line preceding the first stanza is blank.

The text must be divided into stanzas separated by blank lines. Every stanza must have a number that appears at the head of the first line of the stanza, and on the same line.

Although the stanzas need not be entered in the correct order, since they can be reordered automatically, it is necessary for all the versions to be formatted identically: the number of spaces between the left margin and the stanza number and between the latter and the beginning of the line, as well as the number of characters used in the stanza number must be uniform. I describe below (p. 112) how option 2 of the main menu formats the files automatically.

The program package, written in PASCAL, was developed at the IBM Scientific Centers in Madrid and Heidelberg for IBM 43xx computers and more powerful models. It has been subsequently adapted to the UNIX environment of the SUN workstation, a model 3/50 or higher.

Menu environment. The user begins by calling up UNITE’s main menu:

| 1. Operations with files. This option opens the “operations with files” menu: |
| 1. Operations with files. |
| 2. Text Preparation. |
| 3. Identifying the files contained in the texts. |
| 4. Unification in interactive mode. |
| 5. Unification in batch mode. |
| 6. Temporary exit to UNIX. |
| 7. Exit. |

Choose option:

1. OPERATIONS WITH FILES

1. Editing files.
2. Copying files.
3. Formatting the Unified Version.
4. Extracting variants in a file.
5. Working with texts: locating and extracting words, preparing concordances, etc.
9. Main Menu.
Choose option:

1.1 Editing files. This option brings up the “editing files” menu:

1.1 EDITING FILES

1. File of empty and semiempty words. (fich vacuum.scr)
2. File of graphic signs. (fich signs.scr)
3. File of characters that lead to other spelling unification phases. (fich caractgr.scr)
4. File of default execution parameters. (valdefec.dat)
5. File that stores the unified version. (version.txt)
6. File that summarizes the processes of unification. (faclara.res)
7. Others.
9. Main Menu.
Choose option:

1.1.1 Editing the file that contains “empty” and “semiempty” words. Function words with little or no semantic content are used to determine whether or not two lines are wrongly placed inside a stanza, even if they coincide in a number of versions. Agreement with regard to frequently used prepositions, adverbs, or determiners does not in itself mean that two lines are variants of the same verse. What is important for collation is the coincidence of content words. Nevertheless, a parameter decided by the editor fixes the number of coincidences among function words which indicates that two lines are indeed variants of the same verse. Therefore, a list of function or “empty” words cases the tasks of ordering and collating.

1.1.2 Editing the file of graphic signs that will be omitted in the texts when spelling is unified. During the process of collation, the editor may wish to disregard some graphemes, such as punctuation, scribal marks, or mute letters like \( h \).

1.1.3 Editing the file that contains the characters and character strings that lead to other spelling unification phases. UNITE allows the editor to determine which spelling variants can be ignored, or rather equated, in the collation. If,
for example, the difference between -r- and -sr- lacks significance, then this fact must be indicated in the appropriate file. In the Spanish example below (900), the graphs -r-, -sr-, -s, and -s- have all been converted into -s- for the purposes of collation; similarly, -i- represents -i-, -y-, and -y-, while -n- stands for -n-, -y-, and -n-. In consonant clusters, -m- replaces -n- before a bilabial consonant (-mb-/-mp-). Consequently, variants like paso/paso, azor/ azor, sna/sna, campo/campo, etc. will be automatically matched and merged under a single regularized spelling.

1.4 Editing the file that contains default execution parameters. Bymodifying these values, the user is able to control the level of automation of the unified edition. The values contained in this file are the default values when the option for modifying them interactively from the screen is not used.

1.5 Editing the file that stores the unified version. In this option one may edit the product of the collation process.

1.6 Editing the file that summarizes the processes of unification. This file describes how each word created by the unification process is generated by the program, providing the stanza and line where it appears and the manuscript readings that underlie it. It is the source file for the critical apparatus.

1.7 Editing any other file not included in the options given above. It is possible to view the contents of any file included in the package by using the computer-editor (i.e., the visual editor) in UNIX). The user can modify any aspect of the program files as needed, provided only that the notes that appear at the beginning of each file or section are respected.

1.2 Copying files. This option leads to the “copying files” menu, which allows the user to make backup copies of the various files described in 1.1.

Text preparation. This option of the Main Menu permits the user to format the stanzas automatically and to put them in order:

It is advisable to make a backup copy before formatting or ordering a

2. Text Preparation

1. Enter the data format.
   Format by default: blank spaces before stanza number = 2
   length of number = 3
   blank spaces after the number = 2
   lines ignored at the beginning of the file = 4

2. Formatting the text.
3. Ordering the text stanzas.
9. Main Menu.

NOTE: Do not forget that formatting and ordering will be done according to the data introduced in option 2.1.

Choose option:

The program will search for the text files in any directory, provided that the file names include the appropriate path. This ability can be extended to all UNITE utilities which possess file names.

4. Unification in interactive mode. This option carries out the automatic unification, and displays on the screen the partial results of the different unification procedures. It can be helpful for checking the unification of one or several stanzas, but is not advisable when the texts to be unified are long, since the program locks the screen, preventing multi-tasking. The interactive menu consists of the following options:

<table>
<thead>
<tr>
<th>PRIO</th>
<th>NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>text 1</td>
</tr>
<tr>
<td>2.</td>
<td>text 2</td>
</tr>
<tr>
<td>3.</td>
<td>text 3</td>
</tr>
<tr>
<td>4.</td>
<td>text 4</td>
</tr>
<tr>
<td>5.</td>
<td>text 5</td>
</tr>
<tr>
<td>6.</td>
<td>text 6</td>
</tr>
</tbody>
</table>

Do you agree with the data? (s(yes), n(no), q(to quit the program))
4. UNIFICATION IN INTERACTIVE MODE

1. Modifying the execution parameters from the screen.
2. Giving default values to the parameters.
3. Unification.
4. Main Menu.

Note that before executing 4.3 unification, values must be assigned to the parameters by using options 4.1 or 4.2.

5. Unification in batch mode. In batch mode, control is returned to the menu system once the process has started and it is therefore possible to do other tasks simultaneously even if they are outside the UNITE menu. A message from the computer informs the user when execution is finished.

6. Temporary exit to UNIX. This option permits the user to exit temporarily from the UNITE menu environment to the UNIX operating system, or shell, where it is possible to execute any UNIX command. The user returns to the main menu of UNITE by means of the "exit" command.

The Process of unification. All the phases of the unification process are described in this section. Recall that the functioning of these phases is controlled by the values stored in the parameters file or entered into it from the screen (options 1.4 and 4.1 of the menu system). By modifying these values, the user limits the program's capacity for making decisions within the unification process. Moreover, the user can decide whether any given process should be executed and, if so, the range of application.

1. Text reading. The computer reads the text according to the specified format values; the basic unit is the stanza, formed of a varying number of lines never greater than 5. The computer reads the space before the line number, then the line number, and finally the line itself. The stanza numbers are used to determine the correspondences between the different versions. When a line is read, the file containing the function words (1.1.1) is checked, so that only "empty" words occur in the range defined by the parameters, the computer searches for content word correspondences in other lines; if they are found, it means that the lines of one of the versions being compared were mislocated. The selection of the stanza as the basic unit requires more computational power than if only single lines were read and matched against the corresponding lines of the other versions; but it offers the advantage that mislocation of the lines within the stanzas can be automatically detected.

2. Spelling unification. The original spellings are modified into more usual or neutral graphic forms at this stage in order to eliminate, to the extent possible, mere orthographical differences, while obtaining the maximum number of matching words during collation.

Spelling unification is divided into five phases, in each of which a number

of modifications to the original spellings is carried out. These phases are not executed uninterruptedly one after the other; a collation is carried out after the application of each phase so that the original spelling of the texts can be preserved as much as possible. Thus when a word becomes standardized after one of the modifications, it is not further altered by other phases. However, one may also execute a phase called inactive which prevents any alteration of the original spelling and requires that the collation of the texts go forward on that basis.

The five phases of spelling unification are:

An inactive phase which does not modify the orthography in any way.
A phase which suppresses the punctuation marks by eliminating all the characters stored in the punctuation marks file (option 1.1.2 of the menu system).
A phase which substitutes lower case letters for the corresponding upper case ones and non-accented vowels for accented ones.
A phase which standardizes variant characters and character groups; the corresponding modifications are stored in the file accessed through option 1.1.3 of the menu system.
A phase which collapses consecutive identical words into a single word (e.g., "the the" → "the").

It may seem at first glance that some of these phases are too specifically defined to apply to many different types of texts; however, note that the phases which standardize or equate characters (e.g., rewriting g as s in Spanish) and which eliminate the punctuation marks are controlled by the user, who is also able to determine which phases will be executed and in what order. It is thus quite easy to adapt the unification process to the specific characteristics of the texts to be collated.

3. Unification of position for the same word. When the same word occurs in different positions in the collated lines, the program attempts to assign a single position to it according to the number of times it appears in each position in the collated texts and to the relative priority of each version, while verifying that the position chosen does not already have another word assigned to it. The editor can decide whether this process is to be executed, and is free to determine the range of possible positions of each word inside the line, i.e., how far a word can be dislocated from its original position in the text in order to match another. Both the position of the word within the verse and the distance between the variant positions of the word are important parameters.

4. Word-Joining. Two words are joined into a form that can be matched with counterparts in the other collated versions.

5. Word-Splitting. Two parts of a word are separated and each is matched with the corresponding words from other versions.

6. Common Letters. This final phase involves "word formation," i.e., an attempt is made to form a new word from the common letters of words still not unified. Successive collations are carried out with these new words;
the one with the highest positive ratio between the number of common and different letters is chosen.

Before accepting the execution of Word-Joining, Word-Splitting, or Common Letters, a check is made to see if it will yield positive results, i.e., if it will help to choose a new word for the unified version. If it does not, the process is not executed. This check avoids unwanted modifications that might interfere with the results of a later process. The three phases under discussion (4, 5, and 6) work with words located in different positions. Therefore, after accepting the results of a particular process, the program uses the criteria operative in phase 3 (Unification of Position) to determine the location of the words in the unified version.

These three phases are executed both together and as different phases. At first the new words are required to be the same as some already existing words. Later this restriction is relaxed, with the limit that the ratio of identical to different letters must be equal to or greater than 1. This loop is executed whenever there are positions left in the unified version without words assigned to them, and the previous processes give positive results. Thus, "phase 1, 2, or 3" of Common Letters means really "phase n" of the three joint procedures (4, 5, and 6).

When a previous procedure yields positive results, other phases of the same procedure will never be executed consecutively. This ensures that the repeated execution of the same procedure does not prevent other solutions that might be more accurate, as might happen, since the range of tolerance for different spellings becomes broader in each successive phase.

7. The output of the result. The automatic unification process generates two output files. The first stores the unified version, while the second enters data relevant to the three previously mentioned processes (4, 5, and 6) and summarizes their execution. In the file that stores the unified version, the collated texts are divided into stanzas, each separated by a blank line. In addition to the words selected for the unified version, the variants are listed, with a number which identifies the textual witness where they occur. Optionally, the uncollated stanzas of all versions may appear before the unified stanza. The creation of the file that summarizes the process of unification is optional and is controlled by a user-modifiable parameter. Its generation is very useful whenever an explanation is needed for the execution of the aforementioned processes. For each word generated, a line in the file indicates the stanza and line in which it appears, the process that generated it, the unified word resulting from the process, and the original words (accompanied by a label identifying the corresponding version) that generated the unified word. These words appear with their original spellings, not with the standardized forms that the unification processes operate on.

A new version of UNITE is now being developed which will eliminate all limitations with regard to the format of the texts, the maximum number of versions, and of lines per stanza. It is also expected that the final version of UNITE will run on MS-DOS or IBM-OS2 computers.

---

**UNITE in action.** As an example of the application of the UNITE program, I have chosen stanza 51 of the Libro de Alexandre, a Castilian text of the 13th century preserved in a 14th-c. Leonese MS (O) and in a 15th-c. Aragonese or Riojan MS (P), as well as in several minor fragments.

The stanza is preserved in four versions of the poem, identified by the numbers 1, 2, 4 (a printed 18th-c. except), and 5 (= P, O, G, G', resp., in the variants that follow). The following text appears on the screen if the process is executed in the INTERACTIVE MODE (option 4 of the main menu). Certain comments, included here to explain various phases of the process, are placed in parentheses to differentiate them from those that actually appear on the screen while program execution takes place.

(First line of each version)

**Unification of the following lines**

Enpecol Aristiontes co<ne><n>o once bien honrado
Comenc'o don Aristotil cuerno ombr<e> he<n> ietrado
Comenzó Aristiontes, como once bien lenguido
Començ'o Aristiontes como olbre bien lenguido

(In the first unification attempt, the only words accepted are those that appear in the same position in at least three of the four versions collated. The words selected for the unified line appear below in the first line. Un-unified variants appear in the following lines preceded by the number that identifies the witness in which they appear. Note also that the original spellings have been modified in an effort to eliminate graphic differences; e.g., ombr > ombre.)

---

9. Editorially resolved abbreviations are enclosed by angle brackets < >; 'e' stands for <e> with tilde and 'a' for c with cedilla. Since 'e' appears in a printed witness, I did not represent it as <e> in this paper.
Original state of the uniform line

51A Aristóteles como bien

/1enpesol/ /14ome/ /1onrado/
/2omares/2don/2aristorl/2cuemo//2omares/2biene/2leraado/
/4scomero/ /45lenguado/

/sonure/

(In the next step identical words are unified even if they appear in different positions in the collated lines; thus /2biene/ [6th position vs. 5th in the other witnesses] disappears from the apparatus.)

Execution of Unification of Position

51A Aristóteles como bien

/1enpesol/ /14ome/ /1onrado/
/2omares/2don/2aristorl/2cuemo//2omares/2biene/2leraado/
/4scomero/ /45lenguado/

/sonure/

(In the next process, the program attempts to form words from the letters that similar words have in common. The different phases of this process indicate the number of differences between letters that are permitted when forming a unified word. Recall that not only are the number of common letters taken into account, but also the number of versions in which they occur. Hence /2omares/ and /sonure/ only differ in one letter (after b has been rewritten as u for the purposes of collation) but do not yield a unified word because it would correspond to only 2 of the 4 versions collated. However, in Phase = 2 /14ome/ and /sonure/ form om... e since this result corresponds to 3 versions.)

Execution of Common Letters: Phase = 1

51A come...so Aristóteles como bien

/1enpesol/ /14ome/ /1onrado/
/2don/ /2aristorl/2cuemo//2omares/2leraado/
/4scomero/ /45lenguado/

/sonure/

Execution of Common Letters: Phase = 2

51A come...so Aristóteles como om...e bien

/1enpesol/ /1onrado/
/2don//2aristorl/2cuemo//2omares/2leraado/
/4scomero/ /45lenguado/

/sonure/

Execution of Common Letters: Phase = 3

51A come...so Aristóteles como om...e bien le...ado

/1enpesol/ /1onrado/
/2don/ /2aristorl/2cuemo/ /sonure/

(In addition to these processes, Word-Joining and Word-Splitting also can come into play, although they do not apply to the first line of the stanza.)

Computers and Text Editing: Application to Old Spanish Texts

(Second line of each version)

Unification of the following lines

fijo dixol en buena edad soles vos llegado
fijo a buena edad soles embiado
él dio... fijo, a buena edad eres llegado
el dido fijo a buena edad eres llegado

Original state of the uniform line

51B fijo

/12fios/1dixol/1en/1buena/1edad/1soles/1vueso/
/2as/ /2quena/2solas/2enmendid/2sueo/
/45s/ /45fios/45dixol/45fio/45buena/45edad/45sueo/

(buen was rewritten as buena because the program attempted, but failed, to find a coincident reading; buena remains unchanged since it coincides with counterparts in 4, 5.)

Execution of Unification of Position

51B fijo...a buena... llegado

/1dixol/1en/12edad/12soles/1vueso/
/2toral/ /2enendid/
/45s/ /45dixol/45dixo/45enmed/45sueo/

Execution of Common Letters

51B fijo...a... buena... eda... llegado

/12soles/1vueso/
/2quena/ /2enendid/

(3rd line of each version)

Unification of the following lines

pora seyer ome* bueno tu lo as aguisado
de seyer ome buen tu lo bien aguisado
de seyer ome bueno; fizlo bien á guisado,
de seer hombre buen lo bien aguisado

den

Original state of the unified line

51C de ser

/bueno/
/1pasa/1ser/14ome/1tud/12s/1as/1bien/15aguisad/2omares/2tendees/24guisados/45fios/45bien/4as/

(Note how the program unifies ser with ser by collapsing double letters to single letters.)
Execution of Unification of Position

51C de ser bueno bien
/1popor/1isier/14ome/1tu/12leu/1as/15aguaida/ 
/2onure/ /2tenedes/ /24quisaido/ 
/45faslo/ /4a/ 
/Sonure/ 

(In “Word-Joining,” not previously illustrated, certain words are joined if the combined form matches a counterpart in some of the other versions, in this case, /4a/ and /24quisaido/, once joined as /aguaida/, coincide with /15aguaida/. Words are not joined if no positive result is achieved.)

Execution of the Word-Joining: Phase = 1

52C de ser bueno bien aquisaido
/1popor/1isier/14ome/1tu/12leu/1as/ 
/2onure/ /2tenedes/ 
/45faslo/ 
/Sonure/ 

(Word-Splitting, the converse of the previous operation, divides /45faslo/ since /a/ coincides with /12leu/ in /45(fas)/. The parentheses indicate that /f/ is not a word from the original text, but rather has been produced by one of UNITES's processes. Note that /ncl = /nc/ save in the case of /ambro, ambro/, I happened to have a special interest in the phonotactics of this particular word. The program, failing to match /ambro with /ambro/, converts /n/ to /m/ in a still unsuccessful attempt to unify the two forms.)

Execution of Word-Splitting: Phase = 1

51C de ser bueno lo bien aquisaido
/1popor/1isier/14ome/1tu/11o/ 
/2onure/ /2tenedes/ 
/45(fas)/ 
/Sonure/ 

Execution of Common Letters: Phase = 2

51C de ser omue bueno as lo bien aquisaido
/1popor/1isier/11n/ 
/2tenedes/ 
/Sonure/ 

(Fourth line of each version)

Unification of the following lines

sy leuaro lo quereres com<m>o lo as enpecado
sy leuaro lo q<ui>sieres cuemoid /2uaido/compeado
sy leuaro quiesieres como has comenzado
sy leuaro quiesieres como as comenzado

Original state of the unified line

51D si
/12leuaro/111o/1quieres/11comow/11o/1as/1enpecado/ 
/2quisaido/2cuemoid/2auedes/2compeado/ 
/45leuaro/45quisaido/45comenizado/ 

(The program has collapsed /l- to /- and /- to /- in 4 leuaro and 5 leuaro, but still cannot combine them with 1, 2, leuaro because of the word division.)

Execution of Unification of Position

51D si como as
/12leuaro/111o/1quieres/11comow/11o/1enpecado/ 
/2quisaido/2cuemoid/2auedes/2compeado/ 
/45leuaro/45quisaido/45comenizado/ 

Execution of Word-Joining: Phase = 1

51D si leuaro como as
/1quieres/11o/1enpecado/ 
/2quisaido/2cuemoid/2auedes/2compeado/ 
/45quisaido/45comenizado/ 

Execution of Common Letters: Phase = 1

51D si leuaro quiesieres como as
/1quieres/11o/1enpecado/ 
/2cuemoid/2auedes/2compeado/ 
/45comenizado/ 

Execution of Common Letters: Phase = 2

51D si leuaro quiesieres como as co مهمgado
/1quieres/11o/1enpecado/ 
/2cuemoid/2auedes/ 

(The /n in co مهمgado comes from both comenizado and comenizado. The selection of /n/ in this word prevented the correct result comenizado. The program has since been rewritten so that /mpl = /mp/. Once all the lines have been unified, the unified stanza goes to the output file with the following format. Each original line has been rewritten with the words selected in the various phases of the unification process. The variants, still identified as to the witnesses in which they occur, appear below each rewritten line.)

Unified Stanzas

51A como...- Aристотелем como omue bien le المgado
/1enpecado/ /1onrado/ 
/2donu/2aristote2cuemoid/ 
/Sonure/ 

51B fijo dixo...e...a buena eda...- llegó
/12uaido/1uoso/ 
/2uaido/2emuido/ 
/4uaido/
51C de ser om. —c bueno _as lo bien aguisado
                   /1por/1sier/ /1tu/
                   /2tenedes/
                   /5tomar/

51D si leuras quiere es como as co._nsado
                   /1quieres/1lo/ /Lempesado/
                   /2cuenos/2auedos/

This UNIFIED STANZA represents the program's best effort to collapse together all of the text common to the extant versions. The unified version and the variants provide the editor with the material necessary for completing the edition; he now chooses among alternatives on the basis of his knowledge of the author's or the period's usus scribendi and relegates the rejected readings to the apparatus. He may produce a conservative edition, aspiring only to recreate the archetype of the extant texts, as I have done in my edition of the Alexandre (Marcos-Marín 1987a; cf. Greenia 1989), or he may attempt to move beyond the archetype by amending the text ex a peingenii. In my edition I preferred the first solution as more in consonance with my method, which focuses on what the extant manuscripts have in common rather than on what a hypothetical original might have been.

In the edition of the unified text, italics are employed when the two largest MSS, P and O, coincide vis-à-vis the excerptus G and G' (a situation that does not happen to arise in stanza 51), and boldface when one of the long MSS matches an excerpt, but not the other MS. Brackets enclose editorial corrections, and a critical apparatus follows the text.

**51 Come[n]co Aristotel como ombre bien le[pr]ado
fino dixo [il] a buena edat [eres] llegado
de seer ombre bueno [faz]lo bien aguisado
si leuras quiere como as començado**


*Universidad Autónoma de Madrid*
### Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACH</td>
<td>Association for Computers and the Humanities</td>
</tr>
<tr>
<td>ACM</td>
<td>Association for Computing Machinery</td>
</tr>
<tr>
<td>AIOR</td>
<td>Annali Istituto Universitario Orientale, Napoli, Sezione Romanza</td>
</tr>
<tr>
<td>ASNSP</td>
<td>Annali della Scuola Normale Superiore di Pisa: Classe di Lettere e Filosofia</td>
</tr>
<tr>
<td>BAE</td>
<td>Biblioteca de Autores Españoles</td>
</tr>
<tr>
<td>BEXV</td>
<td>Biblioteca española del siglo xv</td>
</tr>
<tr>
<td>BOOST</td>
<td>Bibliography of Old Spanish Texts</td>
</tr>
<tr>
<td>BRAE</td>
<td>Boletín de la Real Academia Española</td>
</tr>
<tr>
<td>CCM</td>
<td>Cahiers de Civilisation Médiévale</td>
</tr>
<tr>
<td>CFMA</td>
<td>Classiques Français du Moyen Age</td>
</tr>
<tr>
<td>CHHum</td>
<td>Computers and the Humanities</td>
</tr>
<tr>
<td>CLHM</td>
<td>Cahiers de linguistique hispanique médiévale</td>
</tr>
<tr>
<td>CN</td>
<td>Cultura Neolatina</td>
</tr>
<tr>
<td>CNRS</td>
<td>Centre National de la Recherche Scientifique</td>
</tr>
<tr>
<td>CRAL</td>
<td>Centre de Recherches et d’Applications Linguistiques</td>
</tr>
<tr>
<td>CSIC</td>
<td>Consejo Superior de Investigaciones Científicas</td>
</tr>
<tr>
<td>DCEC</td>
<td>Diccionario crítico etimológico castellano e hispánico</td>
</tr>
<tr>
<td>DOSL</td>
<td>Dictionary of the Old Spanish Language</td>
</tr>
<tr>
<td>ECr</td>
<td>L’Esprit Créateur</td>
</tr>
<tr>
<td>GRLMA</td>
<td>Grundriss der romanischen Literatur der Mittelalters</td>
</tr>
<tr>
<td>GSLI</td>
<td>Giornale Storico della Letteratura Italiana</td>
</tr>
<tr>
<td>HR</td>
<td>Hispanic Review</td>
</tr>
<tr>
<td>HSM</td>
<td>Hispanic Seminary of Medieval Studies</td>
</tr>
<tr>
<td>Lba</td>
<td>Libro de buen amor</td>
</tr>
<tr>
<td>LLC</td>
<td>Literary and Linguistic Computing</td>
</tr>
<tr>
<td>MedR</td>
<td>Medierro Romanzo</td>
</tr>
<tr>
<td>NCFS</td>
<td>Nouveau Recueil Complet des Fables</td>
</tr>
<tr>
<td>PILFUL</td>
<td>Publications de l’Institut de Lexicologie Française de l’Université de Liège</td>
</tr>
</tbody>
</table>
Bibliography


Bibliography


Tirin: Einaudi.


—. 1940. "The Octosyllabic 'Cuadra Via' of Juan Ruiz." HR 8:125–37, 166–70.


BETTARINI, ROSANNA. 1969a. "Una versione delle 

BIBLIOGRAPHY

Bibliography

---. 1990. Palladius Rutilius Taurus Armentini, Obra de Agricultura, traducida y comentada en 1385 por Ferrer Sastre. Madrid, Wis.: HSMS.
CHRIGG, HOWELL & MARGARET S. WITTMAN, eds. 1987–88. The Medieval Lyric: A Project Supported by The National Endowment for the Humanities and Mount Holyoke College. 3 vols. with cassettes. [South Hadley, Mass.: Mount Holyoke College.]
Bibliography


CONTINI, GIAMFRANCO. 1933. “Il filologo Sartore Deibenetti.” I’Italia Letteraria, 28 maggio, p. 8 [vol.9, no. 22].


1985–. Diccionario etimológico de la lengua catalana. 7 vols. Barcelona: Curial & “La Caixa”.


1984. “Some Microcomputer Programs for Textual Criticism and Editing.”


BIBLIOGRAPHY


MENÉNDEZ PITAL, RAMÓN. 1896. La leyenda de los infantes de Lara. Madrid: Ducalaz.


BIBLIOGRAPHY


MORISCA, GAVI, GASPAR. 1957. "La transmisión de textos y la crítica textual en la antigüedad (I)." "La crítica textual en Bizancio (II)." "Panoramica de la crítica textual contemporánea (IV)." Anales de la Universidad de Murcia. Filología y Letras 38: 3–55; 39: 3–25. [Part III, "La crítica textual desde el Renacimiento hasta Lachmann," was announced but never published.]


MÜLLER, THEODOR, ed. 1878. La Chanson de Roland. 3rd ed. Götingena: Dieter.


