Stylometry
ISSN 2283-8759
pp. 1-41 (2020)

Shakespeare on the Tree (2.0)

Giuliano Pascucci

In the present paper, a phylogeny of Shakespeare's plays has been created following a procedure used in biology to pinpoint filiation or similarity relationships among species or individuals thereof. After explaining the methods and procedures followed, the essay will deal with how the plays are distributed or clustered on the final phylogenetic tree thus obtained. In the last section of this article, a few among the most apparently significant clusters will be taken into account and discussed in order to consider whether they may raise observations, elicit comments, reinforce or debunk any given understanding of Shakespeare's theatrical production. For reasons of space and given the size of the phylogeny yielded by this research, not all the clusters obtained will be analysed. However, the number of examples provided should suffice to show how and to what purposes a phylogeny of Shakespeare's or any other author's works can be used.

1. Using DNA to Build Textual Phylogenies

The attempt to group literary texts in family trees is not new. It is, in fact, the main aim of ecdotics. Moving from "the principle that 'a community of error implies a unity of origin', the critics determine the relations among the extant manuscripts, so as to place them in a family tree" (Canettieri et al. 2005, sec. 1). However, the kind of trees here created are of a different nature. The four examples included in this essay do not represent the history of a single text, rather a number of Shakespearean plays synchronically represented as the leaves at the far end of the trees' branches. As in any other type of phylogeny, all instances grouped into a cluster share a similar degree of kinship.

Unlike computational linguistics methods, which sometimes focus on occurrence, frequency and distribution of terms, Shakespeare's plays are considered here as complex sequences of characters showing patterns that can be extrapolated and investigated, just as well as DNA strings. However, contrary to DNA strings, which only comprise different combinations of the four letters marking nitrogen bases, a literary text is a more complex object and the strings of characters it comprises include spaces between words, punctuation, paragraphs, capital letters and so forth.

This work is inspired by a project developed by Dario Benedetto, Emanuele Caglioti and Vittorio Loreto, researchers at Sapienza University of Rome. In 2002 they presented an automatic procedure meant to solve textual issues such as language recognition, authorship attribution and language classification (Benedetto, Caglioti, and Loreto 2002, 048702). Their method was based on Information Theory and successfully classified texts according to author, language or content. In view of these results, they created phylogenies such as those used in biology to study evolution through filiation, remoteness (similarity) or other types of relationships among species[1].

[1] I have previously used the same method in other works, e.g. in the article "*Double Falsehood*/*Cardenio*: A Case of Authorship Attribution with Computer-

It was not the first time that genetics and linguistics overlapped. A solid interconnection had been established a few years before by David B. Searls in a paper dating back to 1997, in which he wondered, among other things, "whether the techniques used in analyzing other kinds of languages, such as human and computer languages, can in fact be of any use in tackling problems in molecular biology" (Searls 1997, 333).

Nowadays the parallel between the genetic code and language has become intuitive. Expanding the analogy, one could say that the genetic code is the language in which a text is written; DNA is the way in which sentences are arranged and structure the text; genes are sequences of characters whose combination makes a text unique and somehow recognisable. In molecular biology and genetics, remoteness and similarity between species are accounted for by the number of DNA strings they share; the same occurs with texts.

In the field of textual criticism, rare words or *hapax legomena* allow scholars to make meaningful inferences about the texts investigated; however, redundancy is nevertheless essential to discover similarities.

In this light, Maurizio Lana has reinvigorated the analogy between biology and linguistics claiming that style is "the unique combination of genetic elements, namely formal traits, characterising the writings of an author or a corpus of texts […] either generally or at a given time [...]" (Lana 1996, 36, my translation). In the scholar's opinion, redundancy defines style,

---

Based Tools" (Pascucci 2012), in which I addressed the *Cardenio*/*Double Falsehood querelle*, and in "Using Compressibility as a Proxy for Shannon Entropy in the Analysis of *Double Falsehood*" (Pascucci 2017), where the same subject matter was investigated again in detail and including a larger number of plausible Shakespeare's collaborators. Criticism received over time by this method as less performing than Markov chains and Naive Bayesian methods has been commented on by the authors who created the method (Benedetto, Caglioti, and Loreto 2002a, 2002b and 2003). As for criticism received by Shakespearean scholars about how I used the method, in this paper I expand on its details in the hope of clarifying points that may have come across as obscure in the past.

which consists in "the entirety of the criteria which make the communicative model adopted by an author unique and unmistakable" (35, my translation).

According to the above analogy, the present paper illustrates a method to plot Shakespeare's theatrical corpus on a tree-shaped graph. In order to create this graph, the first step is to extrapolate character strings common to different works; then the linguistic remoteness between texts is computed using Benedetto, Caglioti and Loreto's method. The distances thus obtained are subsequently used to create a distance matrix, which will later serve to create phylogenetic trees (see See the Trees appended to the end of this article).

From Darwin onwards, phylogenies have been seminal in the study of evolution. They have proved remarkably accurate in foreseeing viruses' mutations, thereby allowing, for example, exact predictions on what types of influenza one should expect the next year. However, a tree plotting a literary corpus cannot be interpreted in the same way as those accounting for the evolution of animal species. In animals and micro-organisms, changes occur over time when genetic material is vertically transferred, i.e. handed down from a common ancestor to its offspring. In addition, horizontal transfer of genetic material is also possible, for example when a virus hosted in one bacterium penetrates another, thus bringing alien DNA fragments into the new host.

In writing, two mechanisms embody a horizontal transfer. The first is when an author is writing two or more texts at the same time or at almost the same time. In this case the author will probably mark both texts with a few key words, sentences or linguistic patterns either consciously or unconsciously stored in his memory. The second, much more challenging in the present case study, is when two authors pour the above linguistic patterns into a text they are writing in collaboration.

2. Building a Base for Phylogenies

Phylogenetic trees consist in branches joined by nodes, namely taxonomic units.

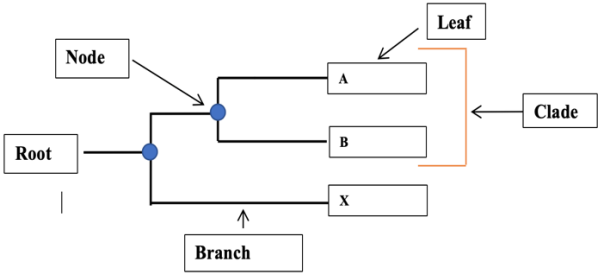


Fig. 1

At the tip of the bifurcating branches springing from each node, there are pairs of leaves representing couples of *taxa*. Only one *taxon* per leaf is allowed. When the nodes follow, or point to, a temporal order, the graph is usually rooted and is defined as a phylogenetic tree. When the graph is only meant to illustrate the relationship between *taxa*, it is unrooted and called a cladogram. In both cases empty leaves are not allowed.

A tree entailing a temporal order, that is a diachronic representation of a text, is particularly suitable when one wants to account for a text's variants, witnesses or collateral manuscripts. Such a tool may lead to the identification or recreation of a common ancestor, namely an Ur-text. Even when there is no common ancestor, a phylogenetic tree will be a paramount tool to illustrate the relationships between the above elements. However, as already mentioned, such research would fall within the scope of stemmatics or ecdotics. One of the main aims of the present paper is instead the synchronic representation of Shakespeare's plays, one that accounts for some degree of similarity or kinship they may bear. As unrooted as they are, cladograms are particularly suitable for this kind of grouping, in that they do not suggest the existence of any single ancestor from which all the others originate.

In other words, cladograms are alien to the metaphysics of origin, and to historical categories such as chronologies. They only illustrate similarities and show pairs of very close relatives on a tree. They are not based on a timeline, nor can they contribute to creating one.

Therefore, even if it is possible to admit that works by the same author may bear some resemblance because they were written around the same time or during a short span of time in which the author's style or linguistic habits had not undergone substantial revision or change, a cladogram will not be able to pinpoint that precise moment in the author's biography.

In terms of genetics, phylogenies are built by measuring the remoteness between two instances, regardless of an original ancestor, whose existence, represented by a root common to all the plotted species, can only be postulated in retrospect (*a posteriori*). The distance between two species or two members of the same species can intuitively be measured counting the differences they show when their DNAs are compared. Once differences have been pointed out and counted, it is possible to create a distance matrix, a numerical representation of such distances, on which the phylogeny will be subsequently built.

For the sake of clarity and brevity, let's analyse five made-up chunks of DNA belonging to the same gene as it appears in five different species: Bonnacon, Parandrus, Monoceros, Hydrus and Crocotta[2].

1)	G T C A T G G T G C T T G	(Bonnacon)
2)	G A T C A A G A G G C C A	(Parandrus)
3)	G T C A T C G T G C G G T	(Monoceros)
4)	G T T C A A A G G G T T G	(Hydrus)
5)	G T G A A A G T G G A T T	(Crocotta)

These are the aligned sequences of the five DNA chunks.

As already mentioned, the first step towards the creation of a cladogram consists in creating a matrix accounting for the differences between species. The process is usually carried out in a pairwise fashion.

In this mock case study, we will start by measuring string 1 and 2, namely Bonnacon and Parandrus. The pair shows ten

[2] In order to avoid misconceptions, I have deliberately mimicked the plotting of DNA resorting to animals commonly described in medieval bestiaries.

differences (highlighted characters) eventually reported on a chart.

1) G T C A T G G T G C T T G (Bonnacon)
2) G A T C A A G A G G C C A (Parandrus)

	BONNACON	PARANDRUS	MONOCEROS	HYDRUS	CROCOTTA
BONNACON		10			
PARANDRUS					
MONOCEROS					
HYDRUS					
CROCOTTA					

Fig. 2

The next step will consist in detecting the differences between sequence 1 and 3:

1) G T C A T G G T G C T T G (Bonnacon)
3) G T C A T C G T G C G G T (Monoceros)

Here it is possible to detect four differences. Again, the number is used to fill out the above chart, which, after this second count, will look like this:

	BONNACON	PARANDRUS	MONOCEROS	HYDRUS	CROCOTTA
BONNACON		10	4		
PARANDRUS					
MONOCEROS					
HYDRUS					
CROCOTTA					

Fig. 3

The procedure is repeated measuring the distance between strings 1-4, 1-5 (respectively Bonnacon-Hydrus, Bonnacon-Crocotta), 2-3, 2-4, 2-5 and so on, until all distances have been measured and the whole chart has been filled.

After all the distances between all the possible combinations of pairs have been computed, the chart will appear as follows:

	BONNACON	PARANDRUS	MONOCEROS	HYDRUS	CROCOTTA
BONNACON		10	4	7	6
PARANDRUS			10	6	8
MONOCEROS				10	6
HYDRUS					6
CROCOTTA					

Fig. 4

Once again, it is necessary to proceed pairwise and observe that in the first line the species showing the least number of differences are those forming the couple Bonnacon-Monoceros. Building the cladogram will therefore begin by representing the proximity of these two *taxa*.

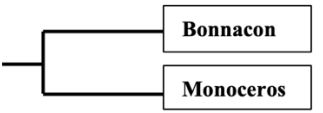


Fig. 5

One rather simple way to proceed in the creation of the cladogram, thus adding new branches, is to create a new chart in which the single specimens just paired are replaced by the pair itself. This allows to compute the average distances between the couple and the remaining specimens. The new chart will therefore appear as follows:

	BONNACON-MONOCEROS	PARANDRUS	HYDRUS	CROCOTTA
BONNACON-MONOCEROS		10	8.5	6
PARANDRUS			...	...
MONOCEROS			...	...
HYDRUS				...
CROCOTTA				

Fig. 6

Because the distance from the Bonnacon to the Parandrus is 10 and the distance from the Monoceros to the Parandrus is once again 10, the average distance between the new couple and the Parandrus will be 10. Eventually, after repeating the procedure

and filling up the chart, one will wind up with a tree like the following.

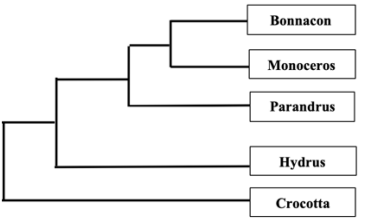


Fig. 7

In the above graph, the relative closeness between the Bonnacon and the Monoceros is visually illustrated and easy to grasp.

Building a tree can be based on two different methods usually defined as distance-based and character-based. The main difference between them is that character-based methods use the aligned sequences directly in the construction of the trees, whereas distance-based methods, one of which has been herein used, first transform the aligned data into distances, then use such values, completely disregarding the initial character sequences. In particular, distance-based procedures can resort to the neighbour-joining method, to a weighted least squares method (Fitch-Margoliash) or to the Unweighted Pair Group Method with Arithmetic mean, also known as UPGMA.

Explaining the theories that lie behind these is not the aim of this paper, nor is the illustration of the mechanisms behind the tree-building algorithms that these approaches utilise[3].

For the purpose of this paper suffice to say that the present research falls within the framework of distance-based methods and that trees have been built at first using the Fitch-Margoliash method, then using the neighbour-joining method, one that

[3] The interested reader will find simple explanations of phylogenies at http://bio1520.biology.gatech.edu/biodiversity/phylogenetic-trees, together with links to other material, including video tutorials on how to build phylogenies.

requires a shorter running time and is therefore best suited for large datasets. Simply put, the algorithm implemented in the neighbour-joining method follows the steps in distance matrix creation described above. Starting from the first pair of closest *taxa*, it creates a node joining them. It then calculates the distance of the rest of the *taxa* from the newly created node, thereby creating a new node and repeating the procedure until all *taxa* have been dealt with.

3. How to Read a Phylogenetic Tree

The reading of a phylogenetic tree starts from its root, if present.

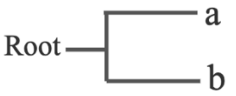


Fig. 8

Reading from the root towards the tips of the tree, where *taxa* (A and B) are located in the above example, means moving forwards in time. The longer the branches the longer the span of time separating an ancestor from its descendants. However, as already mentioned, cladograms are unrooted trees in which the length of branches does not account for the span of time a species needs to spring from a previous one. Their length only depends on the best possible branch disposition found by the tree-building algorithm.

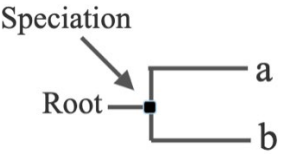


Fig. 9

The point where the branch bifurcation occurs represents a speciation (A and B in Figure 9), the event through which a single ancestral lineage originates two daughter lineages.

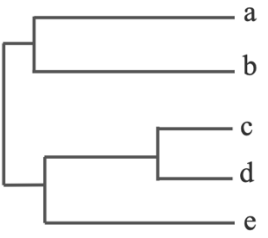


Fig. 10

Figure 10 is an example of a more populated tree. The remoteness between *taxa* is not a notion that can be derived from reading specimens A, B, C, etc. vertically. Trees, cladograms or any other phylogeny can be oriented top to bottom and vice versa or left to right and vice versa. In order to understand remoteness between species or individuals, it is instead necessary to identify lineages. These usually have a history that is partly shared with other specimens, partly unique.

Figure 11 illustrates *taxa* C and D as closely related, although each *taxon* has its own individual development from Z (as is shown by the differently drawn branches). Going further back from Z to Y, their lineages reunite in a common line.

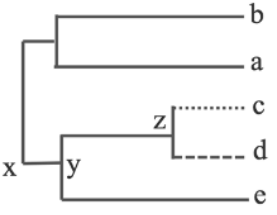


Fig. 11

Contrarily, although D appears equally distant from C and E, D and E are not equally related as C to D. C and D are in fact the offspring of Z, which is not the ancestor of E.

In the reading of an unrooted phylogeny, the position of the bifurcating branches is not meaningful and only follows the tree-drawing strategy of the tree-plotting algorithm. Branch pairs can be rotated 180 degrees leaving unaltered the lineages connecting each *taxon* to its ancestor (see Fig. 12).

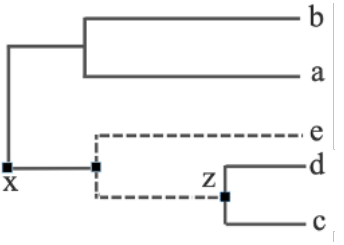


Fig. 12

4. Retrieving Shared Sequences

If one wants to plot texts on a cladogram, it is essential to compute their linguistic distance. What is left once similarities (redundancies) have been removed is merely the sequence of characters that convey the information present in the text. As already mentioned in the previous section, redundancy does not carry information, yet it is important in defining the rules of communication and to ascertain deviations from them. From the point of view of Information Science, repetition implies a non-optimal coding of the message that is being conveyed. Optimal coding intuitively requires the shortest possible sequence of characters, especially for iterated sequences.

This problem was investigated by American engineer Claude Shannon[4] following a stochastic approach, whereas Argentinian

[4] In 1948, while working at AT&T Bell laboratories, Shannon demonstrated that there is a limit to the compressibility of a message. He called the compression limit "entropy", using a term commonly occurring in physics to describe the increasing disorder of a system at its molecular or atomic level. The term was suggested by the mathematician John von Neumann. Up to that moment, Shannon had only formulated the idea of information as "resolved uncertainty". When he asked von Neumann for a better word to call it, the mathematician humorously replied that he had to call it "entropy", not only because information reduces entropy, but also because no one actually knows what entropy is, so in a debate about the subject Shannon would always have the advantage. At the beginning entropy had been a concept only related to the field of thermodynamics. However, later in its history, after von Neumann's suggestion, Austrian physicist Ludwig Boltzmann provided a probabilistic interpretation of entropy "in order to clarify its deep relation with the microscopic structure underlying the macroscopic bodies" (Baronchelli, Caglioti, and Loreto 2005, S70). Although the use of the term may well be

computer scientist Gregory Chaitin, together with Soviet mathematician Andrey Kolmogorov, tackled it logarithmically.

In Shannon's theory, information coincides with how surprising a message is (Shannon 1948a, 379; see also Shannon 1948b). Redundancy can be useful to make sure that a message makes it through the communication channel despite the interference of chance – the noise it may encounter. However, after the first time a string has appeared in a message, its re-occurrence is no longer surprising, i.e. it carries no information, and can be removed from the body of the text. As already mentioned, in Shannon's view there is a limit to how much one can remove in order to downsize a file. He called such limit "entropy". Zipping a message, i.e. computing its entropy, is tantamount to assessing how much information is carried by the message. In other words, in Shannon's view the notions of entropy and information are not only closely related, but even interchangeable.

Gregory Chaitin and Andrey Kolmogorov followed a logarithmic approach to entropy (Chaitin 1969; Kolmogorov 1968). They described the complexity of a digital object as the length of the shortest program that produces the object itself. An example may help clarify their theory. In order to have a computer output the string "AAAAA", one needs a very simple program consisting in one command or instruction: "write capital 'A' five times". However, if the computer must produce a string such as "AGDP134S", the program capable of yielding this output will be much longer than in the previous case: it will have to provide a separate instruction for each character of the sequence. The Chaitin and Kolmogorov definition was therefore also a measure of the resources needed to obtain that output: computers with wide computability resources can afford longer and more complex operations.

---

confusing to scholars of disciplines other than IT, it is intuitive that information provides meaning and structure, thus reducing entropy, which in thermodynamics is the state of disorder to which all systems tend.

Again, Chaitin and Kolmogorov's was a theoretical limit. The ideally shortest program can only be reached by approximation. Zipping is the most suitable procedure to approximate such limit. For reasons of space, this paper will not delve into the mathematics of both approaches. The interested reader will find detailed explanations in the above-mentioned works by Shannon, Chaitin, Kolmogorov, Benedetto, Caglioti and Loreto. What is important is that both approaches look for optimal coding and both reach the same conclusion. No matter the path followed or the point of view from which the issue is tackled, compressors are paramount tools to assess entropy. Removing the unsurprising, iterated chunks of sequences, or reaching the limit of a text entropy, is therefore the precise task that a zipper is expected to perform (Pascucci 2017, 408-9).

For their research, Benedetto, Caglioti and Loreto resorted to LZ77, one of the most common compression algorithms. The modified version of LZ77 they devised was called BCL, the acronym of their surnames.

5. How LZ77 and BCL Work

Abraham Lempel and Jacob Ziv presented their compression algorithm in a paper titled "A Universal Algorithm for Sequential Data Compression" in 1977 (Lempel and Ziv 1977). To compress a file, LZ77 begins to scan it using a sliding window. Compression begins by taking note of each character of the text and goes on until repetitive patterns are found and subsequently stored in a repository called "dictionary". All the sequences in the dictionary are then replaced with a pointer. This contains two figures: the first expresses the distance of a string from the beginning of its previous occurrence, the second indicates its length. Because the algorithm 'learns' and puts aside recursive strings in order to match them with iterations, long texts will yield more iterated patterns, namely wider dictionaries, therefore better compression. The more strings can be removed, the smaller the zipped file. In other words, the longer the text, the more the algorithm will approach the threshold of optimal coding (no waste of characters

for repetitions), a limit that can only be reached if a text has infinite length.

If during the zipping process the typically recurring chunks of characters happen to change, for example due to the use of another language or because of a change in linguistic habits, the algorithm will still be able to compress the file; however, it will need a certain amount of time to learn the new recurring sequences, i.e. to recognise them and begin to store them in the dictionary. During this time, compression would not be as optimal as before the change.

Benedetto, Caglioti and Loreto decided to modify LZ77 so as to take advantage of this limitation. They wanted the algorithm to compress a text using only the patterns learned before the change in linguistic habits, so as to obtain less effective compression. They called the new algorithm BCL. The logic behind their modification was as simple as it was ingenious.

Let's suppose we append a text B to a text A, with A and B having different authors or being written in different languages. When the sliding window of the compressor crosses the A-B junction, BCL will not learn the iterated strings in B. Therefore, compression will not be as effective as when both texts are characterised by the same linguistic patterns. In other words, the compression yielded will not be optimal, because it will be based only on the redundancies characterising A[5].

A feasible and suitable strategy to discover the most similar texts within a repository therefore consists in pairing each text with all the others and zipping the pairs. The best zipping couple will be the one in which text B has the greater number of strings in common with text A. After all the pairs have been compressed it will be also possible to rank the results from the most similar pair to the most dissimilar, thus obtaining a distance matrix on which the final cladogram will be based[6].

---

[5] The idea of appending a text A to a text B in order to compute remoteness had already been suggested in Loewenstern et al. 1995 and in Kukushkina, Polikarpov, and Khmelev 2001.

[6] The distance-matrix algorithm and the neighbour-joining algorithm can be found in PHYLIP, a free package of programs for phylogenies available at

6. A Few Considerations on the Shakespearean Texts Investigated

Modelling is essential to make scientific theories or processes easy to grasp at first sight. In particular, graphical models are essential to visualise a subject as a whole, yet Shakespeare is more ineffable than science and can hardly tolerate this coercion. The scholar trying to graphically represent his plays has to face problems not so different from what other textual scholars had to tackle before the computer era: a complete lack of holographs, which forces us to rely on transcriptions; the presence of sometimes remarkably different coeval versions of the same text circulating among the readers of his time; ensuing ecdotics issues; authorial controversies, multiplicity of spellings, non-normalised use of capital letters, and aberrant verse lineation due to space problems. Maybe Shakespeare's production is already a model, after all. One that has been built over the centuries and that can now provide the best possible approximation to what those texts must have looked like in his time.

This is why procedures such as sequencing and aligning in our case are much more complicated. Which quarto of *Hamlet* is more suitable to carry out a textual experiment? Or wouldn't the Folio version be preferable? Every possible choice is debatable and prone to criticism. In addition, unless the texts needed are entirely rewritten in a machine-readable format, obviously a time-consuming approach, the scholar has to make do with the electronic formats available in the web.

For the present research the texts have been made machine-readable by coding them using ISO Latin-1, an 8-bit character set meant to represent western European languages within Unix-based operating systems and originating from ASCII (American Standard Code for Information Interchange), a standard language used to represent texts in computers. In this encoding each

---

http://evolution.genetics.washington.edu/phylip.html, a webpage by Joe Felsenstein of the Department of Biology at the University of Washington. They have been used within a Unix-based Operating System (Darwin) on a machine equipped with a 2,6 GHz Intel Core i7 6 Core.

character, punctuation mark, space between words, diacritic sign, etc. takes 1 byte.

The texts herein used have been borrowed from a free online website offering a number of Shakespearean resources for students, teachers and academics (www.playshakespeare.com).

1. *Antony and Cleopatra*
2. *As You Like It*
3. *Comedy of Errors*
4. *Coriolanus*
5. *Cymbeline*
6. *Edward 3*
7. *Hamlet*
8. *Henry 4.1*
9. *Henry 4.2*
10. *Henry 5*
11. *Henry 6.2*
12. *Henry 6.3*
13. *Henry 8*
14. *Julius Caesar*
15. *King Lear*
16. *King Richard 2*
17. *Love's Labour's Lost*
18. *Merchant of Venice*
19. *Merry Wives of Windsor*
20. *Midsummer Night's Dream*
21. *Much Ado About Nothing*
22. *Othello*
23. *Richard 3*
24. *Romeo and Juliet*
25. *Taming of The Shrew*
26. *The Tempest*
27. *Troilus and Cressida*
28. *Twelfth Night*
29. *Two Gentlemen of Verona*
30. *Two Noble Kinsmen*
31. *The Winter's Tale*

Literary materials borrowed from the Internet often show inconsistencies. Even within the same repository it is possible to come across erratic usages and standards. From text to text (at times even within the same text), characters' names may appear within brackets or square brackets, they may be capitalised, abbreviated, in italics, etc. Number of acts and scenes may appear in Arabic or Roman numerals and be separated by a comma or a hyphen and so forth. Normalising such chaotic situations may turn out even more time-consuming than rewriting the texts from scratch. Last but not least, electronic texts are entangled with metadata, i.e. the instructions in the markup language used to make the texts available on the Internet (e.g. HTML or XML).

In addition, Shakespearean texts have undergone the attentive sifting of text critics. No matter how accurate additions, emendations, deletions and any other text alterations are, to a computer they are still sequences – a trail of bytes alien to the author.

The most feasible solution therefore consisted in removing punctuation, pilcrows – paragraphs could be the result of space issues in transcriptions rather than a stylistic choice – act and scene indications, stage directions and speech headings. As it will be clarified later, removing the latter was of paramount importance. In order to automatically accomplish these tasks, the author has therefore created a library of scripts[7].

Availability and quality of the available material were not the only parameters affecting the choice of texts to be used. It was also essential to use only completely Shakespearean plays. Critical considerations on collaborative and apocryphal texts have been based on *The New Oxford Shakespeare Critical Reference Edition* (Taylor et al. 2017), presently the most state-of-the-art source of information about authorship issues.

According to the editors of the above critical edition, *Titus Andronicus* is characterised by several authorial hands such as George Peele's and Thomas Middleton's: George Peele, for example, probably wrote the first and possibly the second scene of *Titus*. The so-called 'fly scene' comes across, instead, as a later addition, probably by Thomas Middleton (Taylor et al. 2017, 1:127-28). A similar reasoning applies to *Sir Thomas More*, which Shakespeare only revised, as argued by editor Anna Pruitt (Taylor et al. 2017, 1:1101). Both plays were therefore omitted from this research.

The critical discussion about *Pericles* as a corrupted text casts an ambiguous light on the play. *The New Oxford Shakespeare: Critical Reference Edition* provides a detailed description of a complex authorial scenario (Taylor et al. 2017, 1:1346-47). The impossibility of determining who wrote what demanded that *Pericles* was left out too.

[7] Short programs (series of commands) usually meant to automatically carry out simple tasks operated through a command-line interpreter. In the present work the scripts have been created resorting to Bash (Bourn Again Shell), a Unix Shell and command language, and have subsequently been merged so as to launch a single process. A streamline visual interface has also been created for prospective users.

Timon of Athens is another collaborative play. In it, Middleton's authorial hand was identified long ago and has been more recently confirmed by D. Lake, R. Holdsworth, M. P. Jackson and B. Vickers. However, the play does not seem to be the result of precise labor division. A number of scenes seem indeed written by both authors and show either the inextricable presence of both authors or their alternation (Taylor et al. 2017, 2:3069). Since the risk of including a different author in the experiment was too high, *Timon* was not included in the final repository of plays.

Terri Bourus, editor of *Measure for Measure*, argues that the play was adapted by Middleton before it first appeared in written form, drawing for evidence on various evident additions and deletions. She discusses when the adaptation occurred and which sections of the text were altered. She concludes that "transpositions and deletions are […] difficult and debatable. And it is impossible to be sure about the authorship of smaller passages" (Taylor et al. 2017, 2:1711). These elements seemed reason enough to omit the play.

Orthodox opinion about authorship issues in *All's Well That Ends Well* maintains that the play was written in collaboration with Middleton, yet the extent of such collaboration, although still under investigation, has not thus far produced conclusive results (Taylor and Egan 2017, 278-365). The play was therefore discarded. This is also the case with *Macbeth*, in which the layers of different authorial interventions have forced scholar John Jowett to edit it "as the work of two authors" (Taylor et al. 2017, 2:2999)

On the other hand, in other collaborative plays, the presence of multiple authorship has been verified and their fingerprints better discriminated. Most times it was therefore possible to join all the fragments and have them processed by the algorithms as if they were a whole text. A case in point are some of the histories.

It was therefore possible to include in the experiment Act III of *2 Henry VI* as the only Shakespearean part of the work (Taylor et al. 2017, 2:2471).

As summarised by editor Will Sharpe, partial convergence has been reached on authorship matters in *King Henry VIII*, where the presence of Shakespeare has been unanimously detected only in

the first half of the play, more precisely in I.i, I.ii, II.iii, II.iv and in few other fragments for which, however, the general view is not univocal (Taylor et al. 2017, 2:2746-47). Therefore only I.i, I.ii, II.iii, II.iv have been here included as representatives of the play under investigation.

3 Henry VI deserves separate discussion. The play has recently undergone new investigation carried out by John Burrows and Hugh Craig, who have determined that I.iii to II.ii, II.iv to III.ii, IV.i, V.i, V.iii-vii are Shakespearean, whereas the rest of the play may well have been written by Marlowe (Burrows and Craig 2017, 195). The identified Shakespearean parts have been preserved and used in this experiment.

In her introduction to *1 Henry VI*, Sarah Neville, editor of the text, summarises previous studies that looked for different authors in the text and concludes that Shakespeare only wrote II.iv, IV.ii and some parts of IV.iii-v (Taylor et al. 2017, 2:2387-88). Unfortunately, the size of the text chunk originated by grouping together the three fragments (9 KB) is well below the standard size of chunk for this analysis (32 KB), so it was not included.

A convergence of opinions on attribution issues has been reached about *The Two Noble Kinsmen*. As R. Loughnane says in his introduction, "it is now almost universally accepted" that Shakespeare wrote I.i-iv, II.i, III.i-ii, V.i-iii, V.v-vi. (Taylor et al. 2017, 2:3547), all the other scenes were written by John Fletcher, whereas authorship of the shortest scenes, namely I.v and IV.iii, is still debated. For the purpose of the present experiment all the non-Shakespearean parts of the play and those in doubt have been stripped off.

7. Four Shakespearean Trees

Unfortunately, the BCL algorithm cannot scan entire texts. It has an upper limit of 32 KB. I have already mentioned that in the format used for the experiment (.txt) each character, be it an apostrophe, a space between words or a simple letter, contains a 1-byte piece of information. Conventionally, 1 KB equals 1024 bytes. This means that a 32-KB passage comprises 32,768

characters. This is a substantial chunk of text, but well short of an entire play. The size of Shakespeare's plays varies from 72 KB (*The Comedy of Errors*) to 264 KB (*Cymbeline*).

Each play was split into 32-KB chunks[8]. The remainders of such divisions, if present, were discarded. In the resulting trees, in order to identify chunks belonging to the same play, each fragment was marked by the title of the play followed by two letters. For example, *The Comedy of Errors* size is 72 KB. The division 72/32 KB generated three chunks: *The Comedy of Errors*_aa; *The Comedy of Errors*_ab; *The Comedy of Errors*_ac. Because fragment 'ac', as a remainder of the division, was only 8 KB, it was discarded to avoid the confounding factor of comparing long texts with short ones.

Applied to each of the Shakespearean plays in the available repository, in the first experiment the above procedure generated from one to a maximum of five 32-KB fragments. Because the complete number of chunks was one hundred and the whole number of plays was thirty-one, the number of possible combinations, namely of trees that could be obtained, can be expressed as the product of a sequence of factors:

$$\prod_{i=0}^{n} ai$$

Or:

$$\prod_{i=0}^{31} a_i$$

(where $1 \leq a \leq 5$)

It was therefore rather surprising when, out of 594406696550400 possible combinations, the algorithm picked

[8] This is why, as mentioned before, removing speech headings from the texts was essential. Keeping them would have facilitated the algorithms in acknowledging as similar different chunks of the same play, thus marring the assessment of their efficacy.

(created) one in which each cluster of leaves accurately grouped only chunks belonging to a single play, thus proving remarkably effective in text recognition.

To further test the performance of the algorithm, in the second experiment the stakes were raised and the plays in the Shakespearean corpus were split into 16-KB fragments, which originated one hundred and eighty-nine text blocks[9]. As will be explained in the comment to each tree, the accuracy achieved was once more close to 100%.

The third experiment was instead carried out to test the algorithm's capability in classifying texts according to the language in which they are written. One text in Bokmål (Norwegian) was thus included in the Shakespearean repository as the odd one out: *Et Dukkehjem* (*A Doll's House*). The reasons why this particular play was chosen are as usual related to availability criteria and format parameters. In this case, since the script library designed to automatically remove from the Shakespearean texts' punctuation, stage directions and so on was calibrated on the standards utilised in the Shakespearean corpus, Ibsen's text was processed manually.

The results obtained were once more encouraging: no one of the chunks from *Et Dukkehjem* ended up in one of the clusters of leaves comprising Shakespeare's plays. New experiments were then performed with an increasing number of foreign languages. Every time the results obtained were accurate: all the texts were grouped together according to their language.

The repository on which the fourth experiment in this research is based comprised works by a number of Shakespeare's contemporaries. The aim, here, was to check whether the algorithm could still recompose single works when tackling a number of authors instead of 1 only. In this case a successful grouping would entail author recognition, possibly the most interesting capability of the algorithms, when dealing with

[9] The remainders of the division smaller than 16 KB were discarded as in the previous experiment.

Shakespeare or, more generally, with a system of literary production where authorship is often in doubt.

8. Results

Tree 1

Tree 1 includes Shakespeare works exclusively. To obtain it, the algorithm was fed one hundred text blocks obtained by splitting whole plays into 32 KB, following the limitation of the algorithm sliding window. The new texts thus obtained were presented in the '.txt' format, one in which 1 character equals 1 byte of information. The texts were not in any way recognizable. They were untagged. After processing this overwhelming amount of data, the algorithm created a tree in which each cluster of leaves precisely rebuilds Shakespeare's plays as they were before being split.

Let's look, for example, at *The Comedy of Errors*, which is located near the bottom of tree 1 in the Appendix. After the splitting procedure, the play originated only two text blocks. Picking them out of the one hundred available, the algorithm rebuilt the play laying such blocks at the tip of a bifurcation whose node is marked by number 24[10].

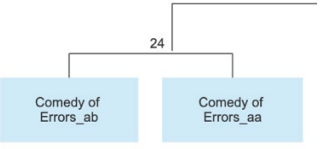


Fig. 13

Another interesting cluster comprises text chunks from *The Taming of The Shrew*. The play, longer than the previous one,

[10] I have here used Google Drawings to illustrate fragments of the phylogenies, which are rendered by the original program FITCH as text files and are therefore not suitable for extracting and printing.

originated three blocks. Since, to obtain the distance matrix, the pairing of texts occurs in a pairwise manner, their final representation must perforce classify one of the fragments as more remote than the others. Cladograms, in fact, only express bifurcating branches. The resulting cluster was therefore as such:

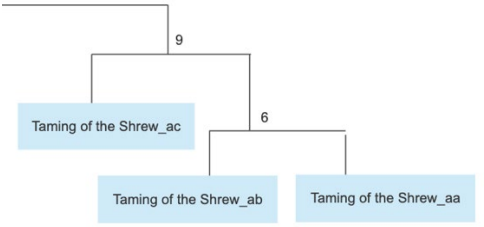


Fig. 14

However, the clustering is still accurate. No pieces of other plays intrude into *The Taming of The Shrew*.

This is even clearer if both clusters are reported together as they appear in the tree.

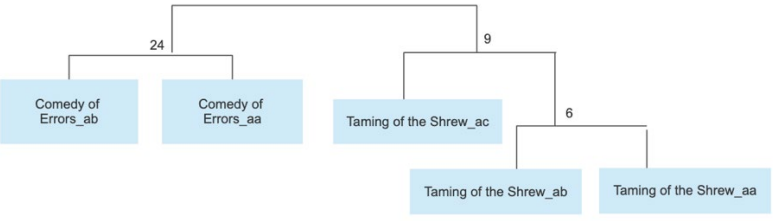


Fig. 15

In this larger fragment it is possible to see that text blocks have been grouped and assigned to a specific cluster according to the play they belong to.

Surprisingly enough, the same phenomenon occurred fairly precisely for all other plays in the whole tree. The following is another example.

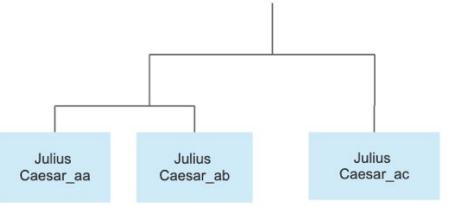


Fig. 16

However, the Roman plays deserve particular attention; in their case, the graph does not limit itself to recomposing them correctly. It also shows that the plays are somehow related by creating a super-cluster of Roman plays, at least as described in 1910 by M. W. MacCallum, who first introduced the expression to designate Shakespeare's plays based on Plutarch, namely *Julius Caesar*, *Antony and Cleopatra* and *Coriolanus* (MacCallum 1910). The following figure is just a part of the whole Roman-play super-cluster, which can be seen in full in the Appendix.

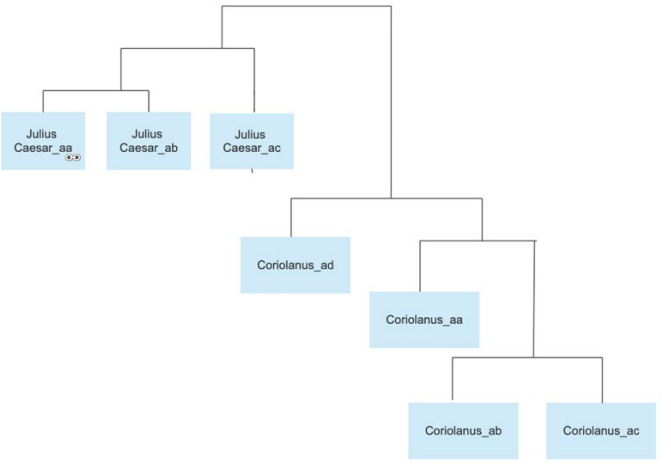


Fig. 17

Super-clusters are therefore another characteristic of the tree, which also groups two among the great Shakespearean tragedies: *Hamlet* and *Othello*. All in all, while performing text recognition and re-creating each single play choosing among all the chunks obtained fragmenting the whole Shakespearean corpus, the

algorithm also seems fairly good at classifying as closely related all the *taxa* belonging to the histories and tell them from those belonging to the tragedies and to the comedies.

This phenomenon needs further investigation. The notions available at present do not allow any inferences on the reasons why the grouping occurs and whether it is possible to increase the algorithm's ability to group texts dealing with analogous subjects even when they do not bear strict resemblance (all the histories narrate events surrounding the lives of English kings, yet this is not reason enough to think that *Richard III* should be linguistically similar to *Henry VIII*, just to give an example).

All in all, when dealing with text recognition, the algorithm has proved almost 100% accurate. The only aberration occurred in the analysis of *2 Henry IV*. Within the huge super-cluster encompassing the histories, *2 Henry IV* is positioned beside the cluster formed by *1 Henry IV* in the following fashion:

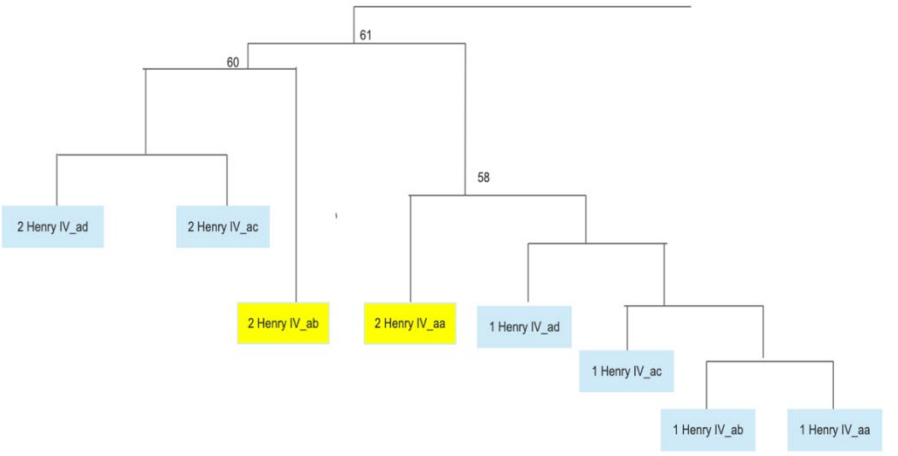


Fig. 18

The graph highlights two fragments of *2 Henry IV* that ended up in two different clusters, one springing off node marked 60 and the other off node 58. This is the only imperfection in the reconstruction of single plays. Here the remoteness indicated by the three nodes differs by a very slim edge, as shown by the figures marking them: 58, 60 and 61. The two text fragments are

therefore very closely related, yet not so much as to be considered by the algorithm as parts of the same play. This may be due, among other reasons, to the well-known problems raised by scene 9 (III.i)[11] and a few other fragments. Authorial inconsistencies about this part of the play have been illustrated by Francis X. Connor (Taylor et al. 2017, 1:761-67). However, the misplacement could simply be due to an error produced by the algorithm or, more likely, by some sort of textual similarity that has not yet been fully disclosed. Even in the case of a trivial error – the remaining fragments from both works have been successfully grouped according to the play to which they belong – the procedure still proves useful. It can at least point out a passage deserving particular attention.

It may be worth pointing out that in this tree the positioning of two fragments of *Antony and Cleopatra* at the top and bottom of the tree is not due to an error. Both fragments are actually located on a bifurcation descending from the same node (the one marked number 10 to be precise) and the fashion in which they are arranged is just a visualization quirk of the software.

Tree 2

Tree 2 was obtained after splitting the Shakespearean corpus into 16-KB blocks in the attempt to ascertain whether the outcome of the algorithm is still accurate when dealing with shorter texts, which intuitively seems a more difficult task. What follows is an example of a much more complex and articulated tree that can be viewed in full in the Appendix. Here, given the increased number of fragments, they have been marked using different colours to make the graph more readable.

[11] Included in fragment *2 Henry IV*_ab.

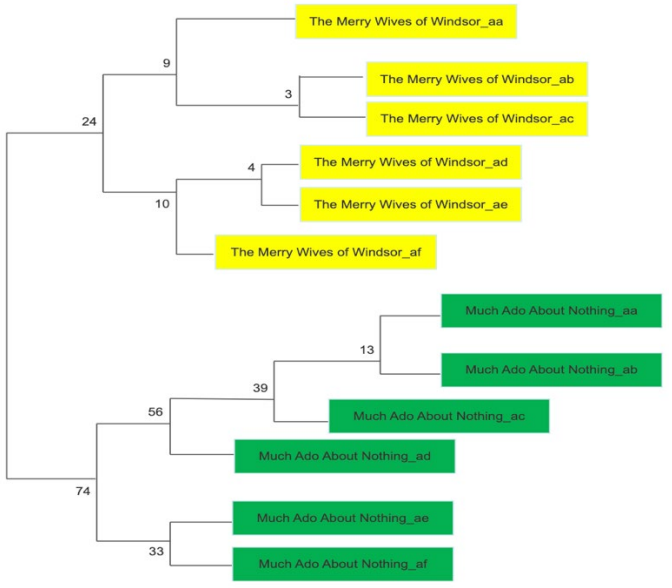


Fig. 19

The results obtained in tree 2 are the same as in tree 1. All fragments have been correctly grouped together despite their larger number and their reduced size. This tree is but a litmus test to tree 1. However, new observations are here in order.

In this case not only can the algorithm cluster the text chunks belonging to the same play. It also seems to have the skill to recognise adjacent sections of the plays, so that it allots a bifurcating branch to the 'aa' and 'ab' pairs of each play, then proceeding to allocate the next ones. We could of course estimate how likely it would be that adjacent fragments would end up in pairs by chance – quite likely if we were dealing with only a handful of cases, but very unlikely when (as here) we have scores of cases.

It has been previously clarified that the cladograms obtained do not account for any timeline whatsoever. Therefore, there should be no connection other than their subject between fragments of the same play. One interpretation of the result would be that logical or cognitive elements linking adjacent sections of a play are pinpointed by the procedure.

The *2 Henry IV* misplacement which emerged in tree 1 here occurred again. This cladogram could therefore be the starting point for further experiments meant to establish whether, decreasing the size of the text chunks, it is possible to delimit the exact parts of texts from which the problem arises or whether the confusion between the two blocks is the result of some 'innate' textual characteristic and cannot consequently be eliminated.

Tree 3

Tree 3 was created including in the Shakespearean corpus a play from a different author in a different language. For this purpose *Et Dukkehjem* (*A Doll's House*) by Henrik Ibsen was chosen because it was available in a form which meant that it could be easily prepared for the experiment. As already mentioned, this preparation had to be performed manually.

The following image describes the portion of tree in which the play appears.

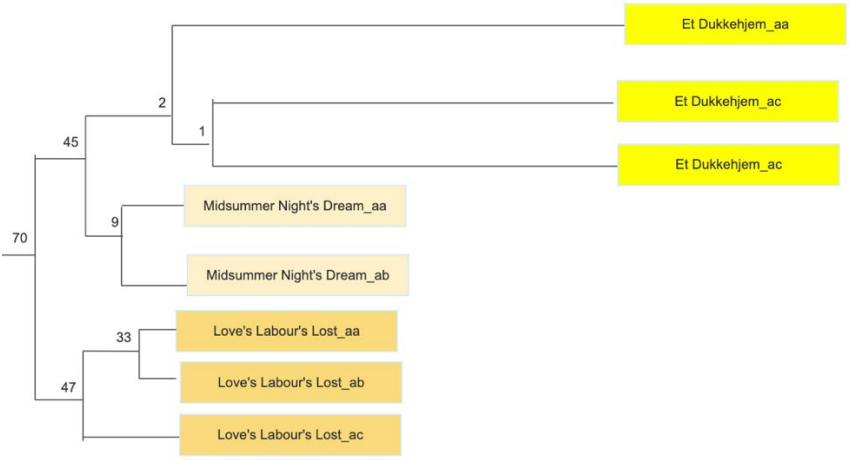


Fig. 20

Enough has already been said about the algorithm's ability to recompose whole plays starting from fragments. The relevant fact here is that the algorithm can work at the same time on different linguistic codes and still perform effectively. No matter what and,

possibly, how many the codes in the corpus are, the procedure can still rebuild complete texts[12].

However, an eye-catching aspect of this cladogram is the length of the branches meant for *Et Dukkehjem*. The procedure seems to consider the play written in the foreign code as the odd one out and isolates it from all the other texts, although not completely: since all files in the repository are zipped with one another, and because all the other texts in this experiment were Shakespearean, the algorithm also computed the distance from *Et Dukkehjem* to all other Shakespeare's works. It is therefore to be expected that, going back to the node from which *Et Dukkehjem* originates and down the other branch springing from the same node, one will find another Shakespearean play, namely *A Midsummer Night's Dream*.

Tree 4

Since the method so far deployed had shown interesting results in text and language, it seemed worth evaluating its capability when tackling authorship attribution. This is an extra step in this research, whose main focus was primarily to create a cladogram of Shakespeare's works both for the beauty of the resulting object per se and to discover whether it could provide new insight on the Shakespearean theatrical corpus of plays.

Tree 4 was therefore built including plays by Shakespeare and by other authors. Again, all texts were divided into 32-KB fragments. The number of texts added is small because of their scarce availability in standards easily convertible into machine-readable formats. Collaborative plays, of which there are many in the period, were excluded since mixed authorship blurs the definition of author and shakes the scientific foundation of the present experiment. The choice was therefore limited to available and certainly non-collaborative plays; from them, the following

[12] The procedure was tested including up to six texts in six different languages. All were successfully recreated. Further research is necessary to test the limit of the procedure. In this case study, the limit could not be further widened because of computational limitations.

have been chosen: *Albovine* and *The Cruel Brother* by William Davenant; *The Bashful Lover* and *A New Way to Pay Old Debts* by Philip Massinger; *Monsieur Thomas* and *Rule a Wife and Have a Wife* by John Fletcher; *Volpone* and *Every Man in His Humour* by Ben Jonson; *The Duchess of Malfi* and *The White Devil* by John Webster. Each play generated 2 text chunks. Only 1 chunk from *The Duchess of Malfi* and *The White Devil* was included in the analysed repository since the second chunks of these plays were too small.

The results here obtained seem encouraging. In the new tree obtained (see Appendix), non-Shakespearean works are grouped together correctly. They appear at the tip of bifurcating branches forming clusters isolated from the rest of the Shakespearean corpus.

Fig. 21

Albovine, for example, is situated at one end of the tree and cannot be confused with any other play. The same applies to *The Cruel Brother*, whose two fragments have been grouped correctly too (see Appendix). However, two clusters come across as particularly interesting.

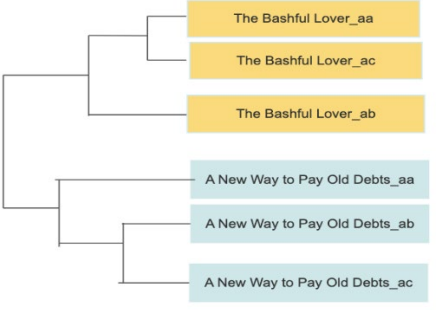


Fig. 22

In the above figure, each cluster (respectively mustard and light blue) accurately recomposes all the fragments of *The Bashful Lover* and of *A New Way to Pay Old Debts*. Yet, these clusters form in their turn a super-cluster encompassing both plays by Massinger. Authorship recognition is here obtained in that all the works by one author are successfully grouped. Indeed, no alien chunks nor alien authors do appear in the super-cluster.

The second cluster worth mentioning is formed by *The Duchess of Malfi* and *The White Devil*. These plays were included to help assess whether the algorithm could detect plays by the same author as well as fragments from the same play. If bifurcations cluster two or more chunks belonging to one text, one might indeed argue that what is being recognised is the text rather than the author. Here the algorithm has clearly grouped the two blocks according to who their author is. There were no other fragments; no complete plays to recompose.

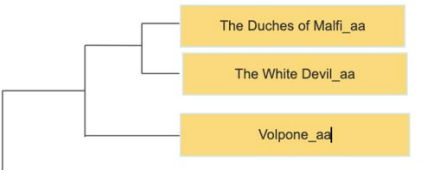


Fig. 23

Again, in this tree, fragments of *Albovine* can be found at the top and the bottom of the tree, yet they do not belong to different clusters. In fact, they spring from the same node (namely the one marked with number 2) and their being at opposite ends of the tree is but a visualisation issue, as is the apparent alphabetical order in which the plays might seem to be arranged (tree 1 has *Antony and Cleopatra* at its top and tree 4 has *Albovine*). Titles of the plays have been removed together with the previously mentioned textual elements and the software utilised does not include any sorting algorithms.

9. Afterword

The procedure illustrated in this paper is agnostic. It requires no previous knowledge of the subject matter treated nor of its language. If repeated, the experiments it entails yield consistent results. The outcome of each test does not change even when experiments are performed by a different human operator on different computers.

This is because, unlike human readers, the sliding window of the zipper scans the texts in search of repeated symbols, not of meaningful language units. Therefore, the sequences of characters it retrieves seldom match with words, phrases or whole sentences. On the contrary, they are usually quite short scraps including spaces between words, diacritics, if present, and word fragments. The few examples illustrated in the following table are meant to convey a sense of what redundant series of characters look like. The small array presented also includes a few rare longer sequences (words and phrases) to provide a general view of the strings in BCL's dictionaries.

Line	*Edward III*	Line	*Romeo and Juliet*
1552	Artois and all look underneath thy	1224	look upon thy death
1601	Darby Ill look upon the Countess mind	874	a was a merry man took up the child
2023	be gone and look unto your charge	5082	revive look up or I will die with thee
1584	the king is in his closet malcontent	1166	drums in his ear at which he starts
1552	Artois and all look underneath thy	329	Where, underneath the grove of sycamore
1555	Undoubtedly then some thing is amiss	763	Compare her face with some that I shall
1872	Scour to New-haven some there stay for	3028	By my head here come the Capulets
1550	till after dinner none should interrupt him	2569	to dinner thither
			I'll to dinner hie you to the cell
			mourners and stay dinner
1158	acquaint me with your cause of discontent	954	And see how one another lends content
1458	the king is in his closet, malcontent	4037	I am content so thou wilt have it so
1281	O that I were a honey gathering bee	1698	O that I were a glove upon that hand

Meaning is alien to most of the strings and, more generally speaking, to the whole procedure described. Zipping algorithms process texts and detect redundancy. It goes without saying that, when referred to the algorithm sliding window, the term "reading" is just a figurative expression bearing little resemblance to the common reading process. Normally we read semantically and semiotically, whereas the machine examines characters one by one while taking note of their position. This is why the results obtained by Benedetto, Caglioti and Loreto were surprising. Is it really possible to automatically classify the content of a text using a tool that completely disregards meaning? The answer given by the Italian scientists is 'yes'; however, although founded on a reliable scientific base, their reply comes across as no less surprising. Can the fingerprints of an author, or the subject treated in a text, wind up entangled in meaningless sequences of characters? Yet this is not the only question the procedure raises. Is text recognition in readers indeed related to content and meaning recognition? Given two fragments of a text in which character names, punctuation, paragraphs and all paratextual elements have been stripped out, will a human reader still be able to recognise their common origin? Can the placement of texts on a tree suggest ideas that scholars may later on confirm resorting to more traditional textual analysis tools?

In the third section of this paper, I have explained that the positioning of the branches is neither significant nor meaningful. Two clusters may be close on the tree, yet if they do not trace back to the same node, their similarity is very little. This is because all the specimens that will go to make the eventual cladogram are added one by one. When a pair forms, a new branch is created, then the algorithm jumbles all the branches available until the optimal positioning for all of them has been found. The result of this methodology is not the 'real' tree, one in which all similarities are represented, rather the best possible approximation to what one would ideally expect. In other words, the more one goes back to previous nodes, thus including a progressively larger number of clusters, the more the degree of similarity among the clusters decreases. Therefore, assumptions based on clusters originating

from distant nodes, even though represented as adjacent on the tree, should be taken cautiously.

However, disregarding the nodes from which the clusters spring and simply reading the 32-KB Shakespearean tree from top to bottom, new and more complex, though dangerous, interpretive possibilities seem to open up. For brevity's sake I will limit their discussion to just a couple of them.

I have so far noted only cases in which clusters are related to the obvious category of genre, particularly emphasizing the Roman plays.

Let's observe the same cluster together with the adjacent ones. Reading the tree vertically the first three clusters, here reported in the form of a list for reasons of space, are:

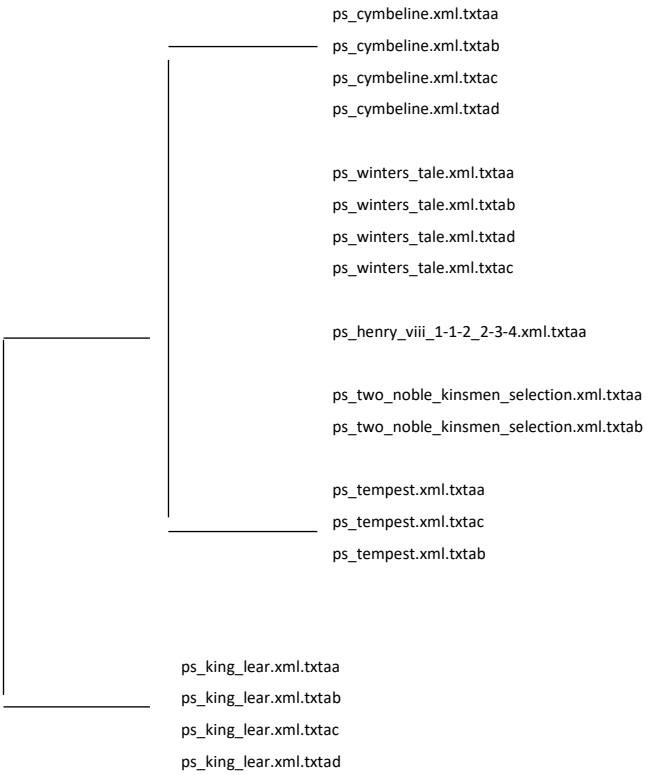
antony_cleopatra.xml.txtab
antony_cleopatra.xml.txtac
antony_cleopatra.xml.txtad
antony_cleopatra.xml.txtaa

julius_caesar.xml.txtaa
julius_caesar.xml.txtab
julius_caesar.xml.txtac

coriolanus.xml.txtaa
coriolanus.xml.txtab
coriolanus.xml.txtac
coriolanus.xml.txtad

Adjacent to *Coriolanus*, though springing from a completely different and distant branch is *Cymbeline*, one of Shakespeare's late plays. Cymbeline's speeches and actions are manifestly rooted in his Roman bringing-up and inspired by Roman values such as honour and courage, possibly in the attempt to recreate a new Roman Empire in Britain. "One might also mention that common to the Roman plays is a focus on military exploits, with the accompanying tumult, confusion, and occasional exercise of magnanimity" (Bergeron 1980, 31). All these elements, which led David M. Bergeron to define *Cymbeline* as the "last Roman play", also make it the most suitable candidate to be positioned next to

the real Roman plays on the cladogram. The cluster comprising *Cymbeline* looks like this:



Again, for reasons of space, the real tree with all the bifurcations leading to the above sample has not been reproduced. This super-cluster encompasses plays characterised by different contents and settings, thus corroborating the idea that some similarity, either in style or content, should rather be expected not only among the leaves included in each single cluster, but also among nearby clusters. However, one cannot refrain from noticing that the grouping, here, is formed by Shakespeare's late plays. Is the compression process responding to those elusive elements that Russ McDonald defined as "the distinctive properties discernible in the late verse […] intimately related to the shift from tragedy to romance" (McDonald 2006, 44)? In this

light the positioning of *King Lear* on the same bifurcation as the late plays seems to suggest some kind of resemblance or analogy between this play and the late works included in the super-cluster, which scholars may choose to explore from a readerly rather than computational perspective.

The last idiosyncrasy I will mention is the peculiar positioning of *Romeo and Juliet* at the bottom of a huge cluster of comedies, springing off branch marked number 90 on the cladogram and adjacent to the cluster comprising the histories. We can speculate about how this play might bridge the gap between the comedies and the darker and more tragic atmosphere of the histories. *Romeo and Juliet*'s deviation from the tragic genre is actually evident. The love language used by the two lovers, the bawdy talk of the nurse, the general atmosphere of the play from the beginning to the moment of Tybalt's murder, and the use of music are all elements more commonly found in comedy. Even the final death of the protagonists comes across as the consequence of a more comedic twist of chance than as the result of the fate usually looming over tragedy. Analysis of the compression procedure itself can offer no light on these matters, but so much of the patterning of the clusters corresponds to familiar groupings like genre that exceptions invite explanations in terms of the content of the fragments.

10. Conclusions

While the present research was being carried out, a number of unexpected results were shown by the trees that were being built. The more the research developed, the more the trees came across as promising in different areas of text analysis. The earlier idea on which this research was based was the creation of a phylogeny representing the theatrical corpus of Shakespeare's plays. The limitation in the number of characters that the sliding window of the compression algorithm could read imposed the splitting of texts into a number of fragments: a serendipity which brought about a decisive change. Realising that the tree-building algorithm could recreate a text starting from its fragments became therefore

the foundation of this attempt to prove how phylogenies can be useful in literary studies.

Using BCL and the neighbour-joining algorithm to build cladograms proved almost 100% correct in text classification and recognition. Authorship attribution, in the light of phylogenetic-tree construction, seemed accurate too, but it certainly needs a larger number of experiments on known authors to better understand its reliability. For reasons of space, time and computability, in this paper it was only possible to suggest how these tools may be used and to what extent they allow inferences.

During preliminary experiments meant to test the effectiveness of the procedure thus far described, attempts made on literary texts written in the eighteenth and nineteenth centuries proved even more effective. However, Elizabethan and Jacobean texts are a much more slippery ground than the more modern texts in print. Turning them into a machine-readable format without altering their nature is a long and complicated process which requires not only deep carefulness, but also a profound knowledge of the whole historical and literary scene at the end of the sixteenth century and the beginning of the seventeenth century. Most importantly, the scarcity of definitively attributed plays and the prevalence of collaboration mean that authorial attribution is likely to remain a highly controversial field, and one where new tools, approaching the language of the texts in an unexpected manner, like the one presented in this paper, are worth considering at least, as part of the armoury of the attributionist.

References

Baronchelli, Andrea, Emanuele Caglioti, and Vittorio Loreto. 2005. "Measuring Complexity with Zippers." *European Journal of Physics* 26, no. 5 (September): S69-S77. https://doi.org/10.1088/0143-0807/26/5/S08.

Benedetto, Dario, Emanuele Caglioti, and Vittorio Loreto. 2002a. "Language Trees and Zipping." *Physical Review Letters* 88, no. 4 (January): 048702.

https://doi.org/10.1103/PhysRevLett.88.048702.

Benedetto, Dario, Emanuele Caglioti, and Vittorio Loreto. 2002b. "On J. Goodman's Comment to 'Language Trees and Zipping'." Preprint, submitted 13 March, 2002. https://arxiv.org/abs/cond-mat/0203275.

Benedetto, Dario, Emanuele Caglioti, and Vittorio Loreto. 2003. "Benedetto, Caglioti and Loreto Reply." *Physical Review Letters* 90, no. 8 (February): 089804. https://doi.org/10.1103/PhysRevLett.90.089804.

Bergeron, David M. 1980. "*Cymbeline*: Shakespeare's Last Roman Play." *Shakespeare Quarterly* 31, no. 1 (Spring): 31-41. https://doi.org/10.2307/2869367.

Burrows, John, and Hugh Craig. 2017. "The Joker in the Pack?: Marlowe, Kyd and the Co-authorship of *Henry VI, Part 3*." In Taylor and Egan 2017, 194-217.

Canettieri, Paolo, Vittorio Loreto, Marta Rovetta, and Giovanna Santini. 2005. "Ecdotics and Information Theory." *Rivista di Filologia Cognitiva* 3 (December). http://filologiacognitiva.let.uniroma1.it/ecdotica.html.

Chaitin, J. Gregory. 1969. "On the Length of Programs for Computing Finite Binary Sequences: Statistical Considerations." *Journal of the ACM* 16, no. 1 (January): 145-59.

Kolmogorov, N. A. 1968. "Three Approaches to the Quantitative Definition of Information." *International Journal of Computer Mathematics* 2, nos. 1-4: 157-68. http://dx.doi.org/10.1080/00207166808803030.

Kukushkina, O. V., A. A. Polikarpov, and D. V. Khmelev. 2001. "Using Literal and Grammatical Statistics for Authorship Attribution." *Problems of Information Transmission* 37, no. 2 (April): 172-84. https://doi.org/10.1023/A:1010478226705.

Lana, Maurizio. 1996. "Testi, stile, frequenze." In *Lingua, letteratura, computer*, edited by Mario Ricciardi, 30-45. Torino: Bollati Boringhieri.

Lempel, Abraham, and Jacob Ziv. 1977. "A Universal Algorithm for Sequential Data Compression." *IEEE Transactions on Information Theory* 23, no. 3 (May): 337-43. https://doi.org/10.1109/TIT.1977.1055714.

Loewenstern, David, Haym Hirsh, Peter Yianilos, and Michiel Noordewier. 1995. "DNA Sequence Classification Using Compression-Based Induction." *DIMACS Technical Report* (April). https://doi.org/doi:10.7282/T3SJ1Q2Q.

MacCallum, M. W. 1910. *Shakespeare's Roman Plays and their Background*. London: Macmillan.

McDonald, Russ. 2006. *Shakespeare's Late Style*. New York: Cambridge University Press.

Pascucci, Giuliano. 2012. "*Double Falsehood*/*Cardenio*: A Case of Authorship Attribution with Computer-Based Tools." *Memoria di Shakespeare* 8 (2012): 351-72. https://doi.org/10.1400/210326.

Pascucci, Giuliano. 2017. "Using Compressibility as a Proxy for Shannon Entropy in the Analysis of *Double Falsehood*." In Taylor and Egan 2017, 407-16.

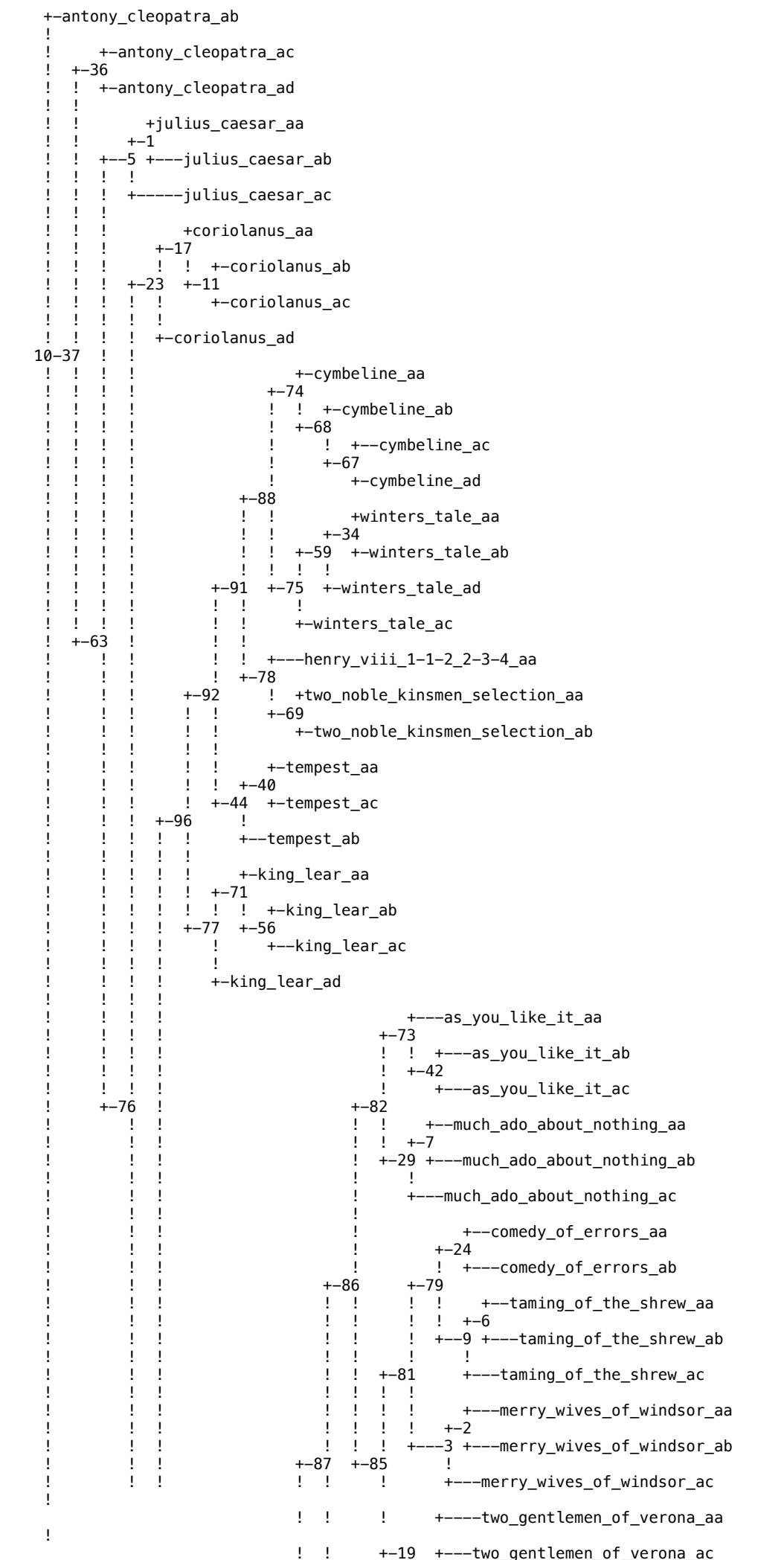
Searls, David B. 1997. "Linguistic Approaches to Biological Sequences." *Cabios Invited Review* 13, no. 4 (August): 333-44. https://doi.org/10.1093/bioinformatics/13.4.333.

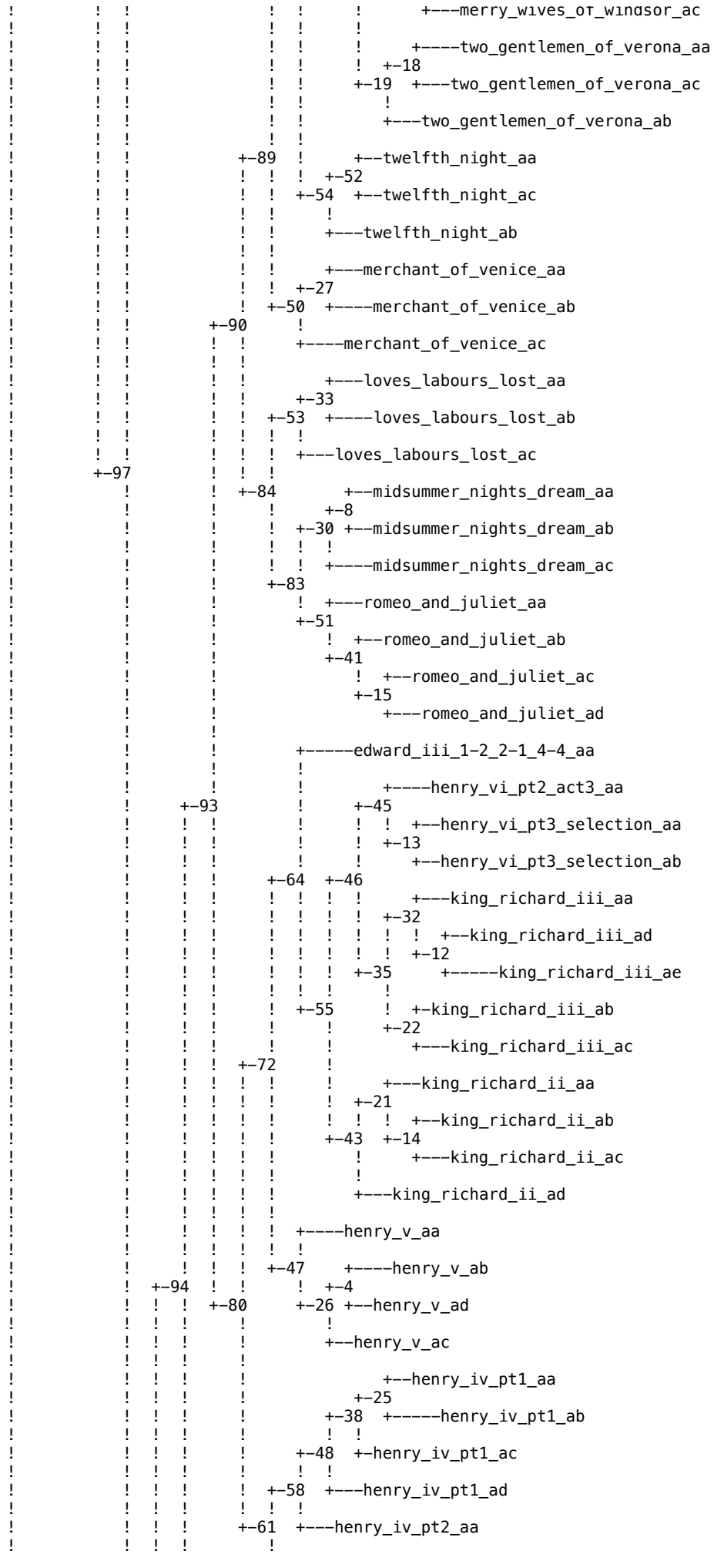
Shannon, C. Elwood. 1948a. "A Mathematical Theory of Communication." *The Bell System Technical Journal* 27, no. 3 (July): 379-423. https://doi.org/10.1002/j.1538-7305.1948.tb01338.x.

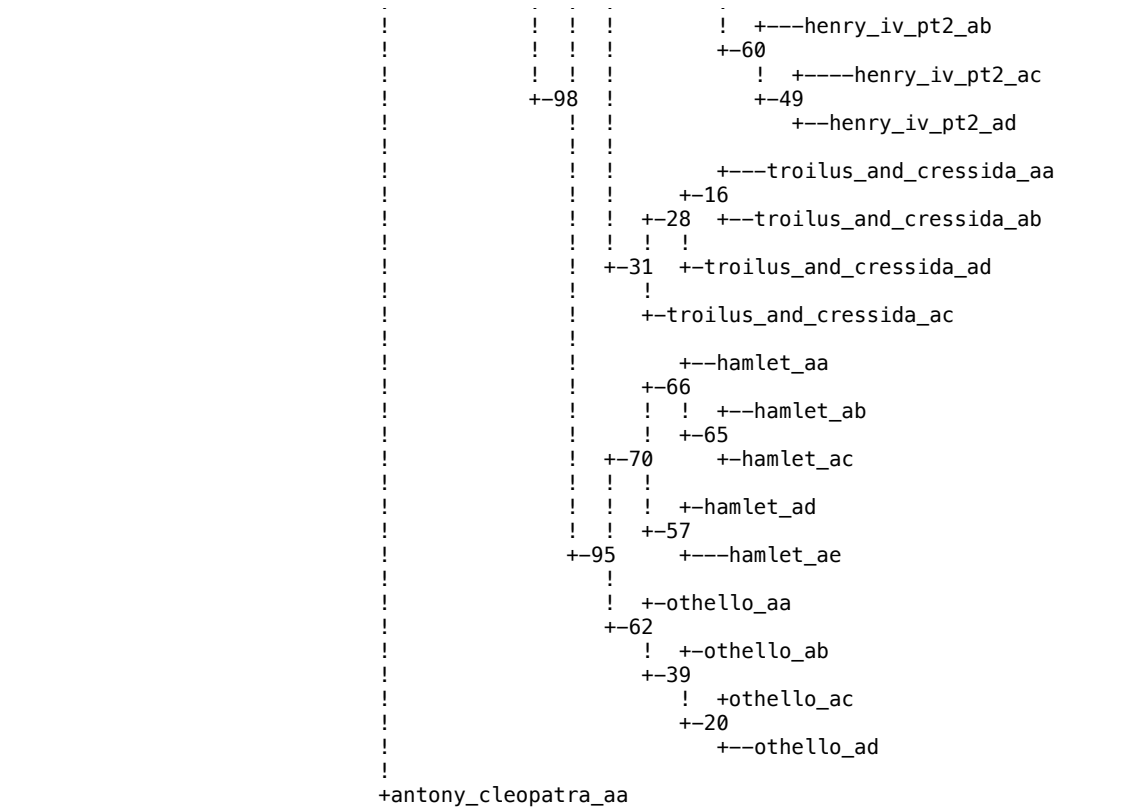
Shannon, C. Elwood. 1948b. "A Mathematical Theory of Communication." *The Bell System Technical Journal* 27, no. 4 (October): 623-56. https://doi.org/10.1002/j.1538-7305.1948.tb00917.x.

Taylor, Gary, and Gabriel Egan, eds. 2017. *The New Oxford Shakespeare: Authorship Companion*. Oxford: Oxford University Press.

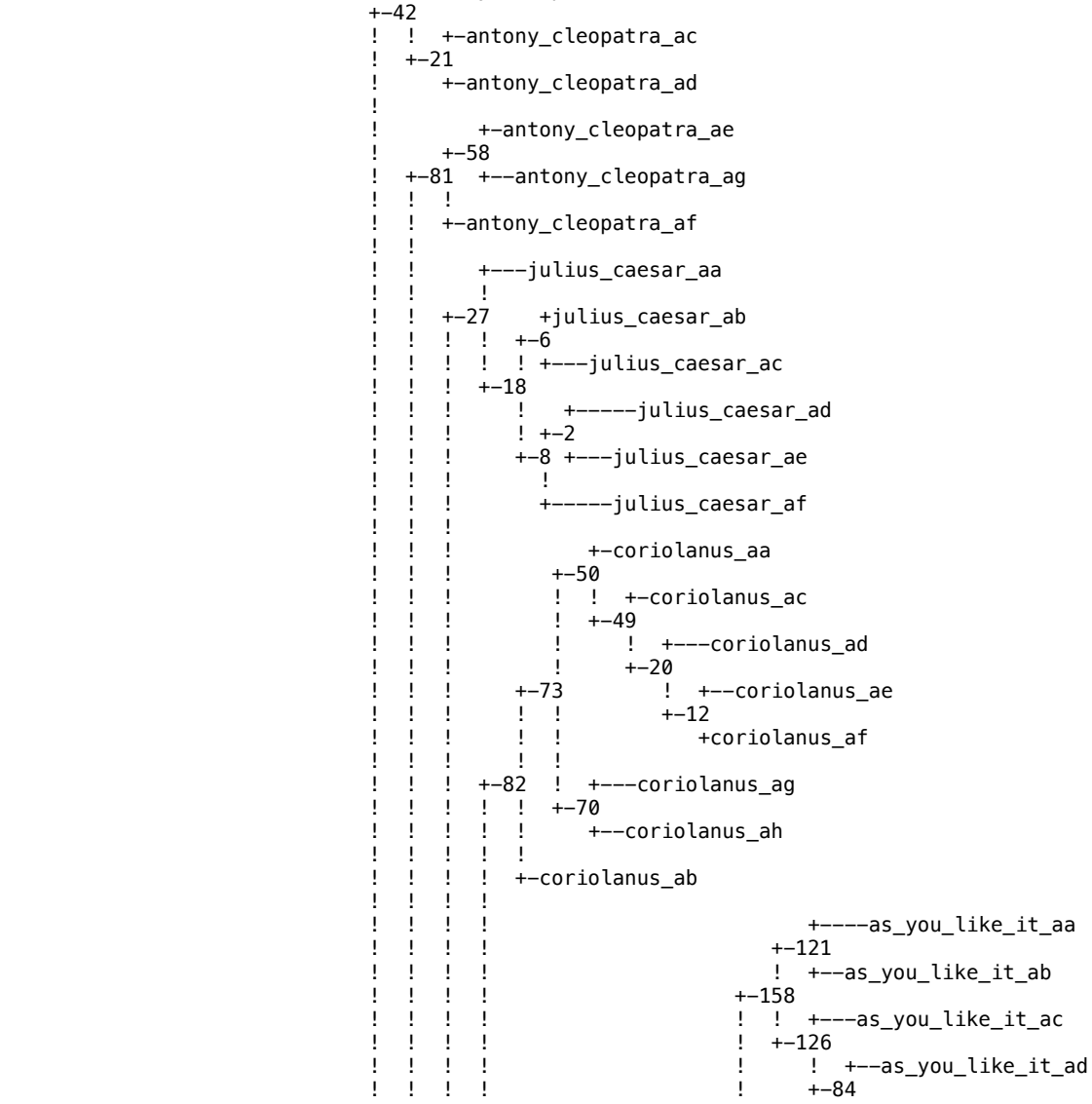
Taylor, Gary, John Jowett, Terri Bourus, and Gabriel Egan eds. 2017. *The New Oxford Shakespeare: The Complete Works: Critical Reference Edition*. 2 vols. Oxford: Oxford University Press.

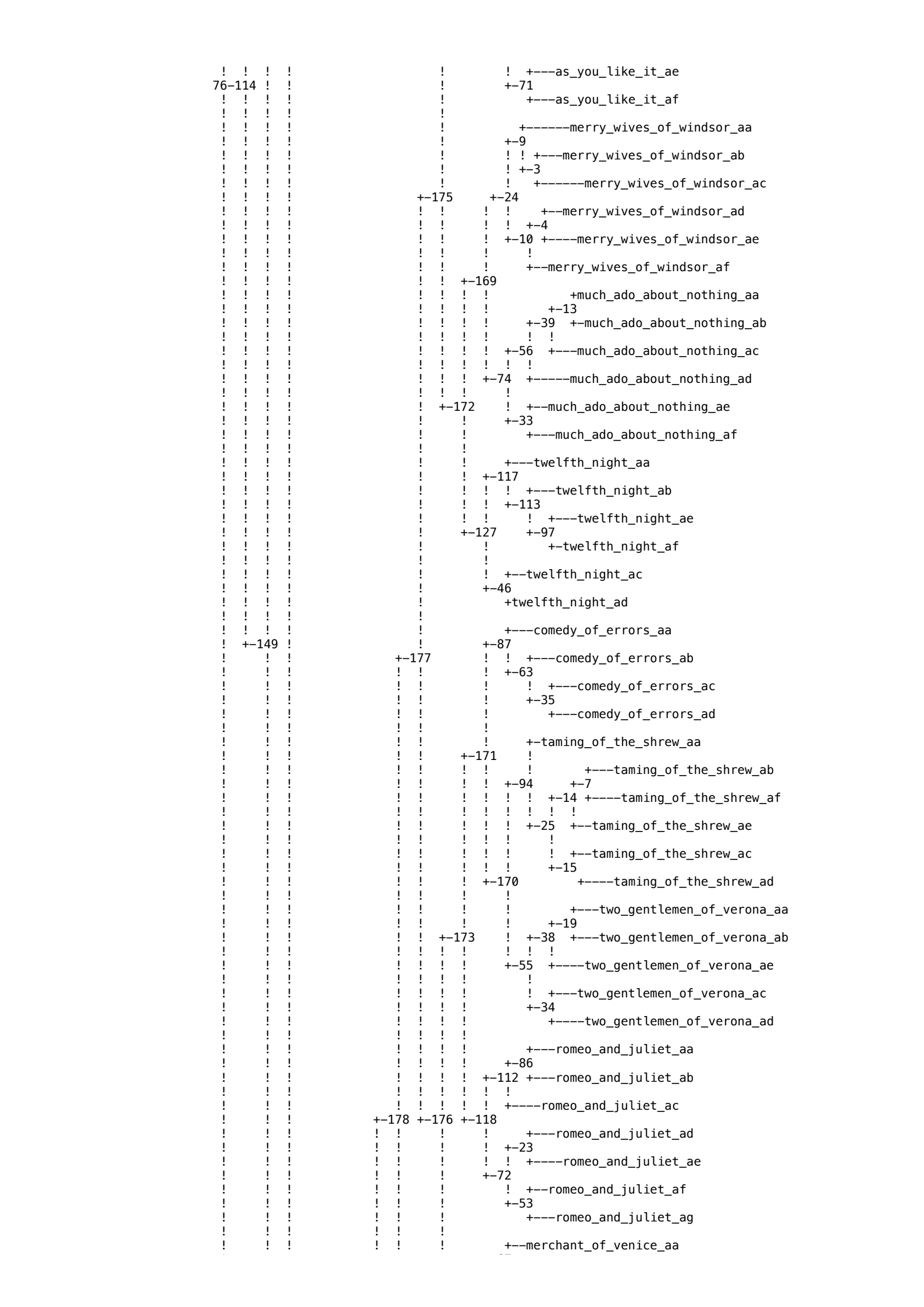

antony_cleopatra_ab
antony_cleopatra_ac
antony_cleopatra_ad
julius_caesar_aa
julius_caesar_ab
julius_caesar_ac
coriolanus_aa
coriolanus_ab
coriolanus_ac
coriolanus_ad
cymbeline_aa
cymbeline_ab
cymbeline_ac
cymbeline_ad
winters_tale_aa
winters_tale_ab
winters_tale_ad
winters_tale_ac
henry_viii_1-1-2_2-3-4_aa
two_noble_kinsmen_selection_aa
two_noble_kinsmen_selection_ab
tempest_aa
tempest_ac
tempest_ab
king_lear_aa
king_lear_ab
king_lear_ac
king_lear_ad
as_you_like_it_aa
as_you_like_it_ab
as_you_like_it_ac
much_ado_about_nothing_aa
much_ado_about_nothing_ab
much_ado_about_nothing_ac
comedy_of_errors_aa
comedy_of_errors_ab
taming_of_the_shrew_aa
taming_of_the_shrew_ab
taming_of_the_shrew_ac
merry_wives_of_windsor_aa
merry_wives_of_windsor_ab
merry_wives_of_windsor_ac
two_gentlemen_of_verona_aa
two_gentlemen_of_verona_ac

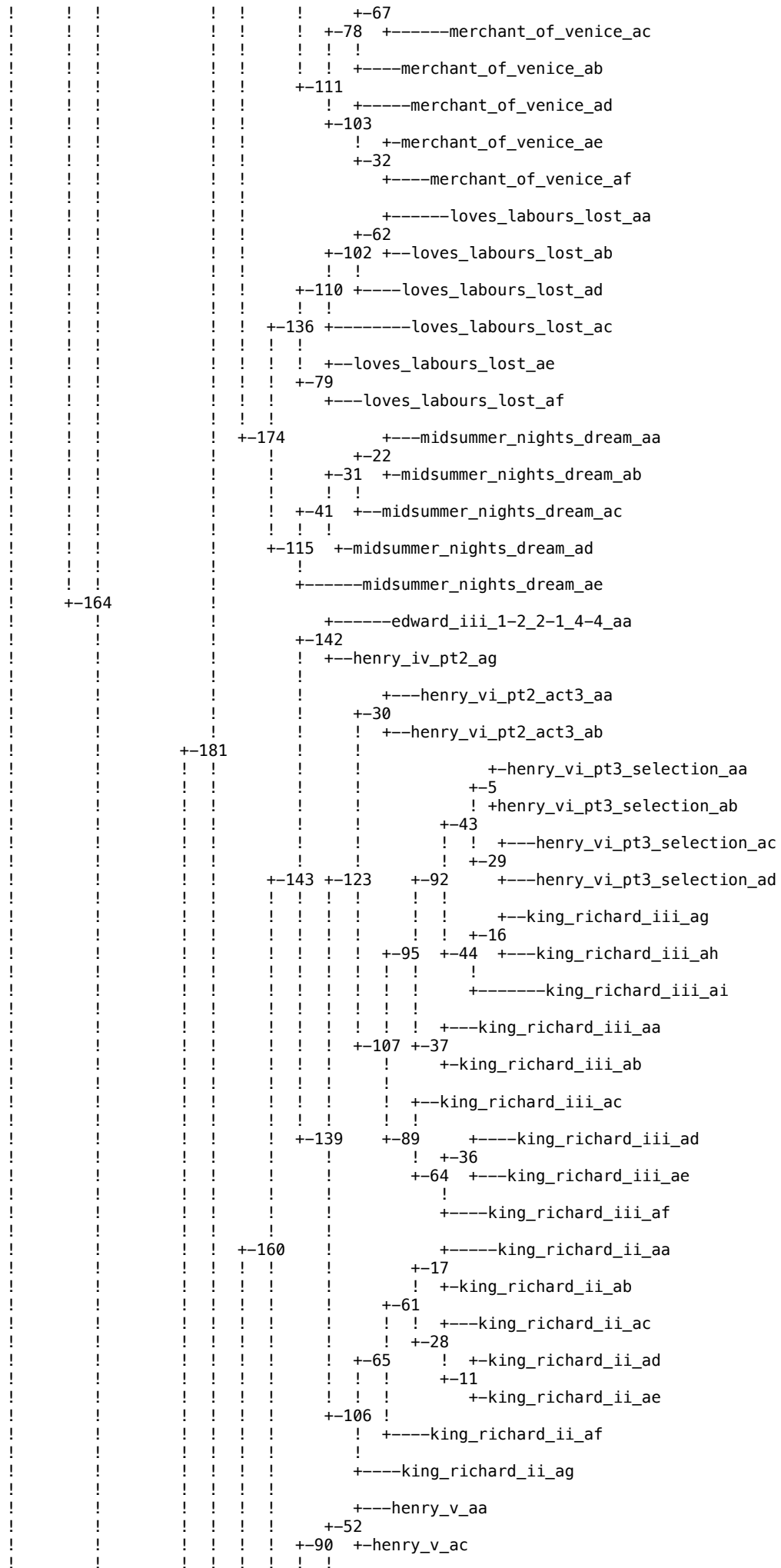

merry_wives_of_windsor_ac
two_gentlemen_of_verona_aa
two_gentlemen_of_verona_ac
two_gentlemen_of_verona_ab
twelfth_night_aa
twelfth_night_ac
twelfth_night_ab
merchant_of_venice_aa
merchant_of_venice_ab
merchant_of_venice_ac
loves_labours_lost_aa
loves_labours_lost_ab
loves_labours_lost_ac
midsummer_nights_dream_aa
midsummer_nights_dream_ab
midsummer_nights_dream_ac
romeo_and_juliet_aa
romeo_and_juliet_ab
romeo_and_juliet_ac
romeo_and_juliet_ad
edward_iii_1-2_2-1_4-4_aa
henry_vi_pt2_act3_aa
henry_vi_pt3_selection_aa
henry_vi_pt3_selection_ab
king_richard_iii_aa
king_richard_iii_ad
king_richard_iii_ae
king_richard_iii_ab
king_richard_iii_ac
king_richard_ii_aa
king_richard_ii_ab
king_richard_ii_ac
king_richard_ii_ad
henry_v_aa
henry_v_ab
henry_v_ad
henry_v_ac
henry_iv_pt1_aa
henry_iv_pt1_ab
henry_iv_pt1_ac
henry_iv_pt1_ad
henry_iv_pt2_aa

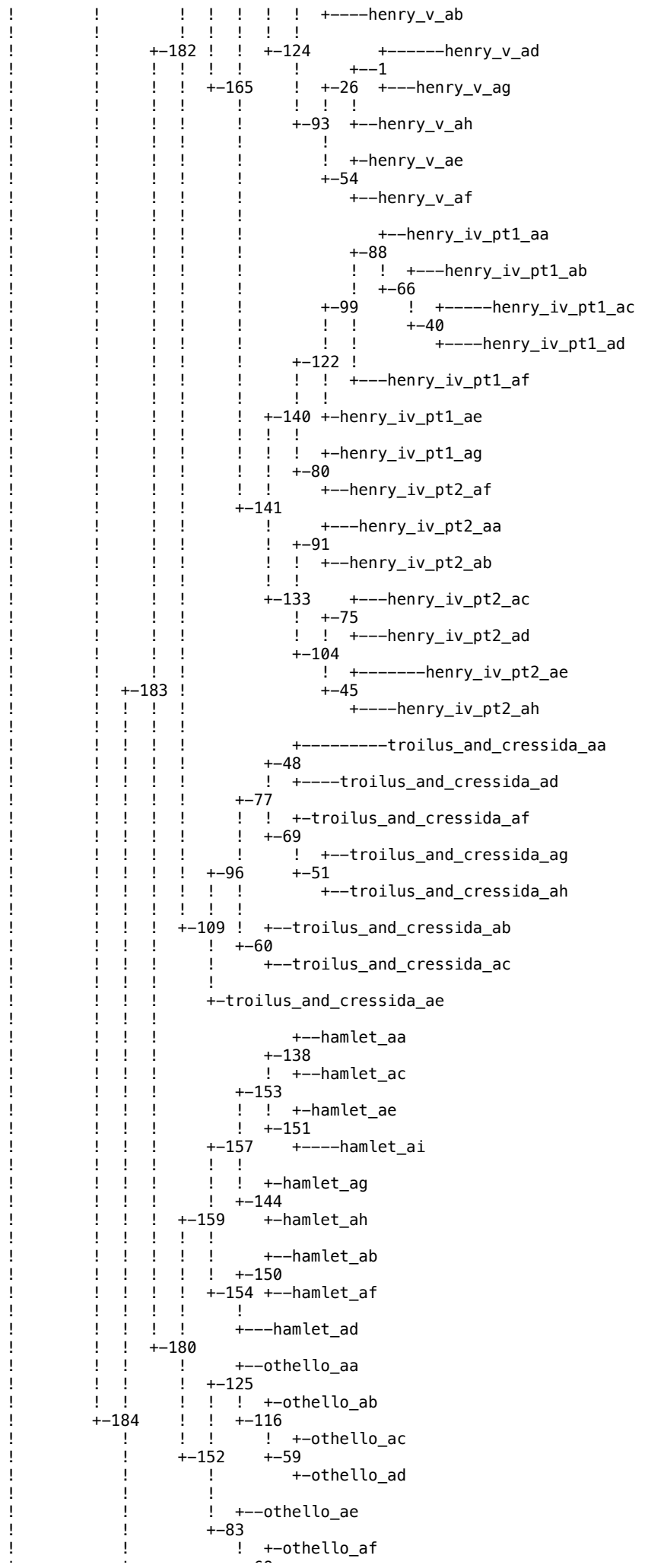


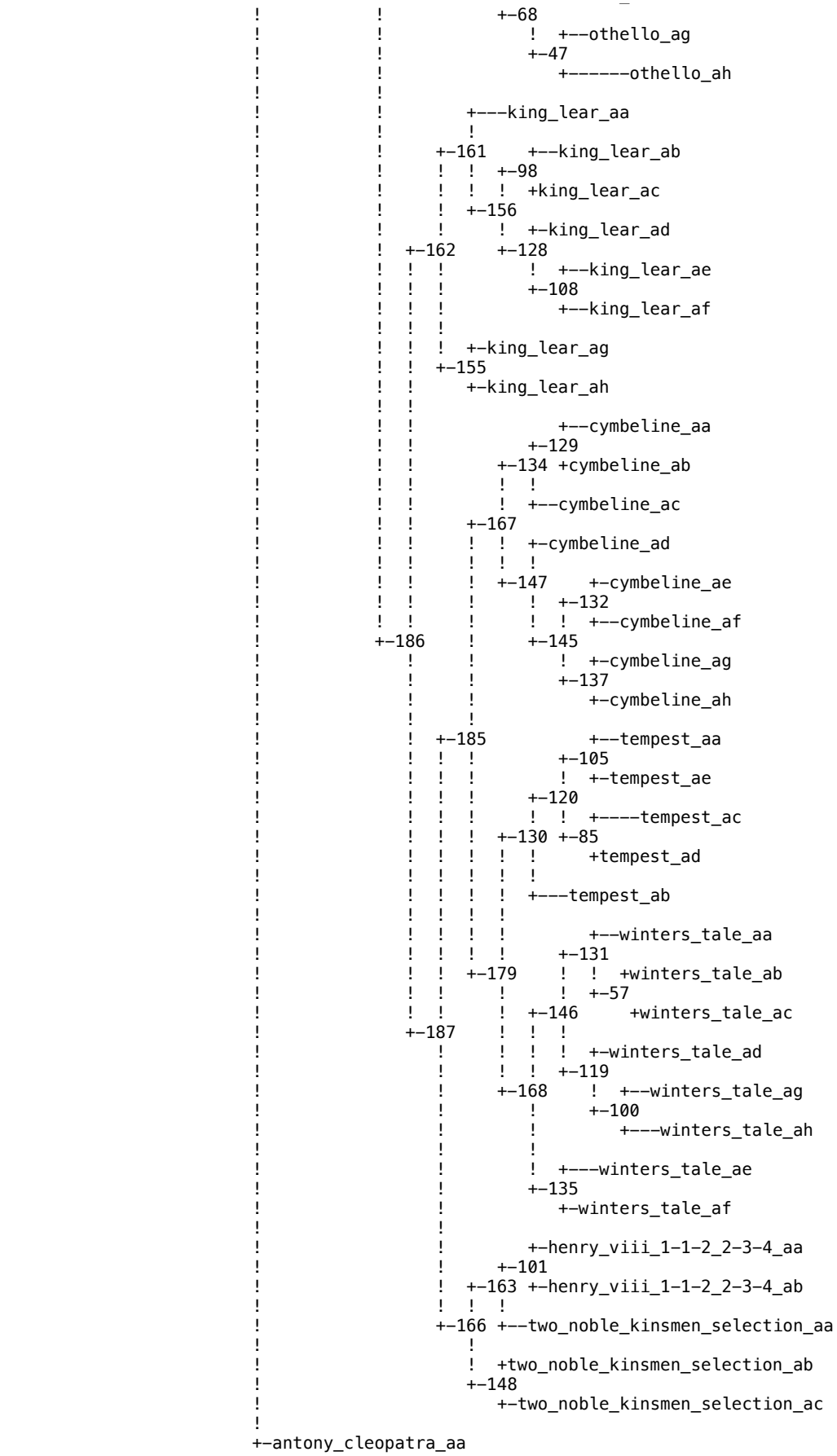
TREE 2 – SHAKESPEARE ONLY (16KB)



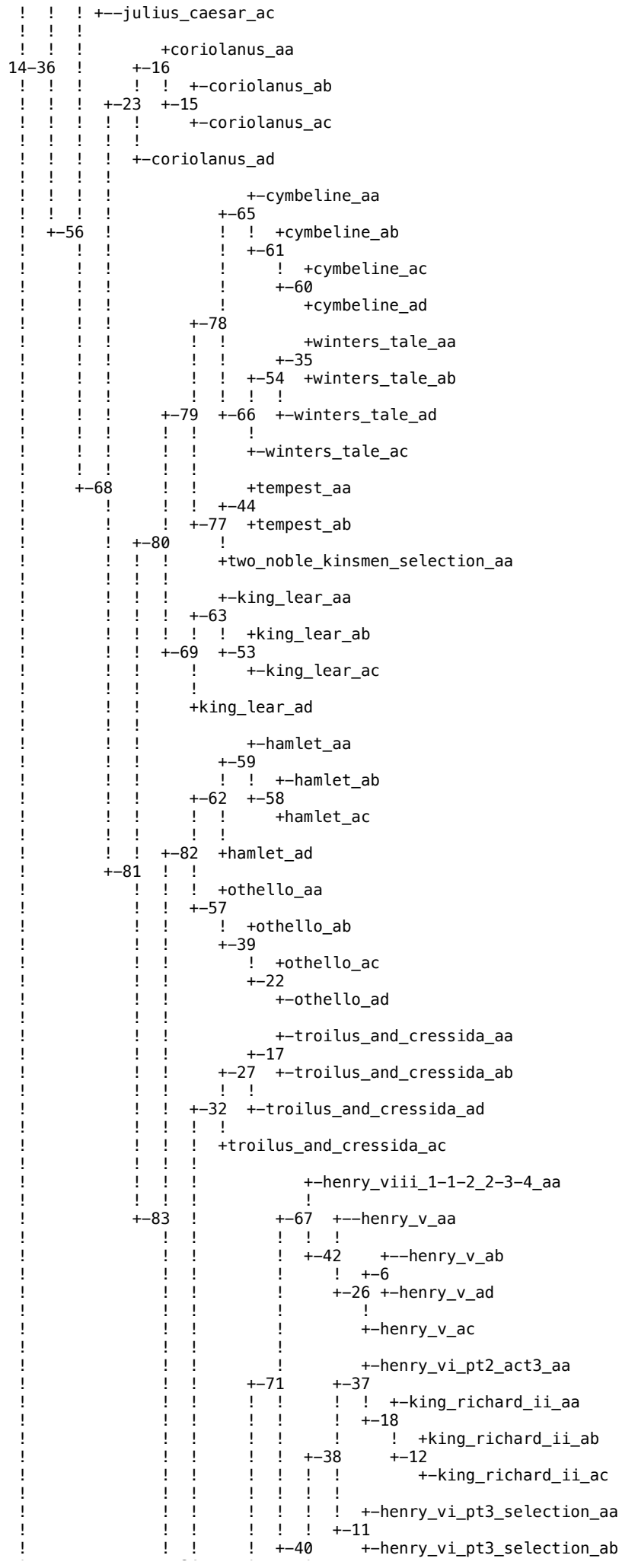

as_you_like_it_ae
as_you_like_it_af
merry_wives_of_windsor_aa
merry_wives_of_windsor_ab
merry_wives_of_windsor_ac
merry_wives_of_windsor_ad
merry_wives_of_windsor_ae
merry_wives_of_windsor_af
much_ado_about_nothing_aa
much_ado_about_nothing_ab
much_ado_about_nothing_ac
much_ado_about_nothing_ad
much_ado_about_nothing_ae
much_ado_about_nothing_af
twelfth_night_aa
twelfth_night_ab
twelfth_night_ae
twelfth_night_af
twelfth_night_ac
twelfth_night_ad
comedy_of_errors_aa
comedy_of_errors_ab
comedy_of_errors_ac
comedy_of_errors_ad
taming_of_the_shrew_aa
taming_of_the_shrew_ab
taming_of_the_shrew_af
taming_of_the_shrew_ae
taming_of_the_shrew_ac
taming_of_the_shrew_ad
two_gentlemen_of_verona_aa
two_gentlemen_of_verona_ab
two_gentlemen_of_verona_ae
two_gentlemen_of_verona_ac
two_gentlemen_of_verona_ad
romeo_and_juliet_aa
romeo_and_juliet_ab
romeo_and_juliet_ac
romeo_and_juliet_ad
romeo_and_juliet_ae
romeo_and_juliet_af
romeo_and_juliet_ag
merchant_of_venice_aa

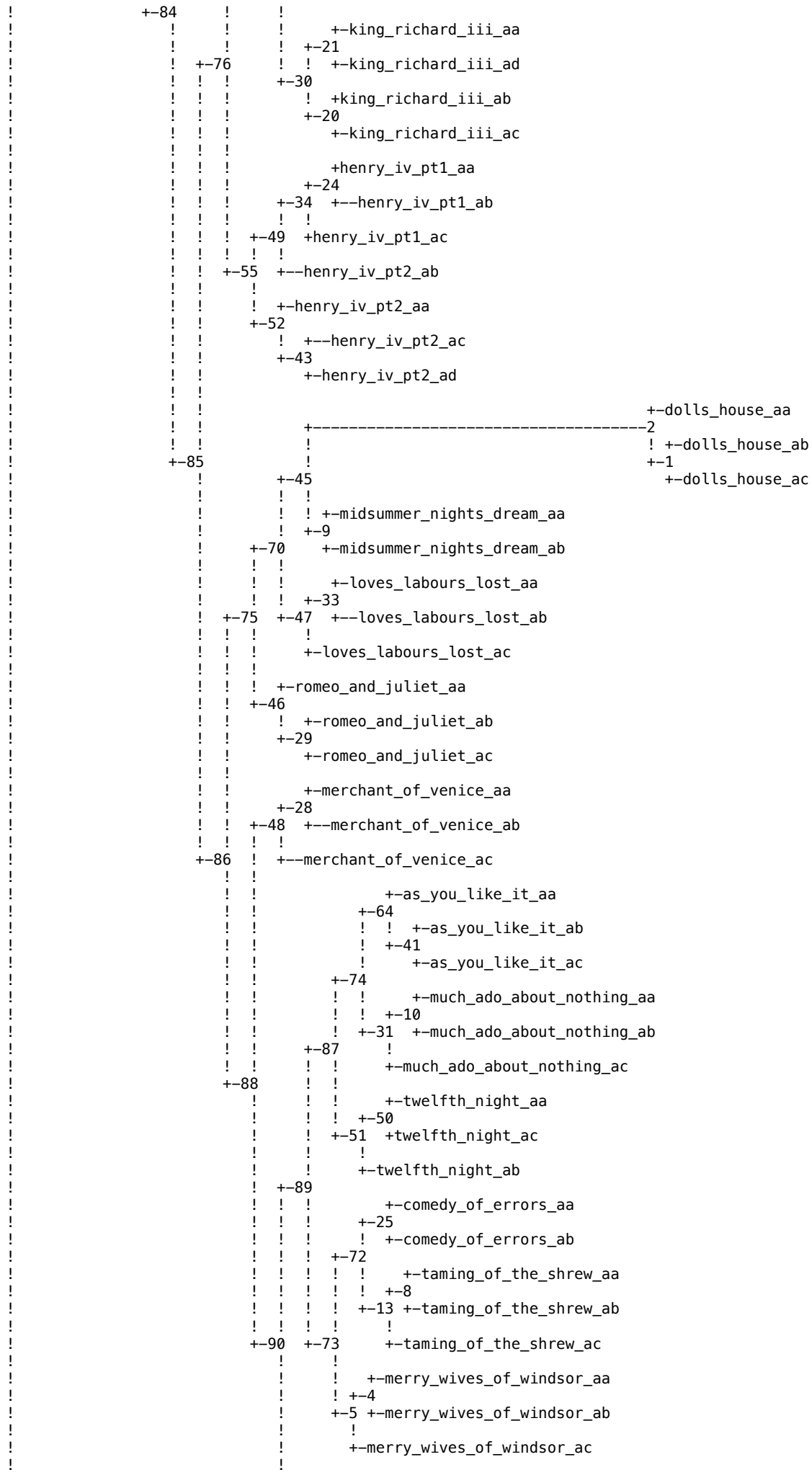

merchant_of_venice_ac
merchant_of_venice_ab
merchant_of_venice_ad
merchant_of_venice_ae
merchant_of_venice_af
loves_labours_lost_aa
loves_labours_lost_ab
loves_labours_lost_ad
loves_labours_lost_ac
loves_labours_lost_ae
loves_labours_lost_af
midsummer_nights_dream_aa
midsummer_nights_dream_ab
midsummer_nights_dream_ac
midsummer_nights_dream_ad
midsummer_nights_dream_ae
edward_iii_1-2_2-1_4-4_aa
henry_iv_pt2_ag
henry_vi_pt2_act3_aa
henry_vi_pt2_act3_ab
henry_vi_pt3_selection_aa
henry_vi_pt3_selection_ab
henry_vi_pt3_selection_ac
henry_vi_pt3_selection_ad
king_richard_iii_ag
king_richard_iii_ah
king_richard_iii_ai
king_richard_iii_aa
king_richard_iii_ab
king_richard_iii_ac
king_richard_iii_ad
king_richard_iii_ae
king_richard_iii_af
king_richard_ii_aa
king_richard_ii_ab
king_richard_ii_ac
king_richard_ii_ad
king_richard_ii_ae
king_richard_ii_af
king_richard_ii_ag
henry_v_aa
henry_v_ac

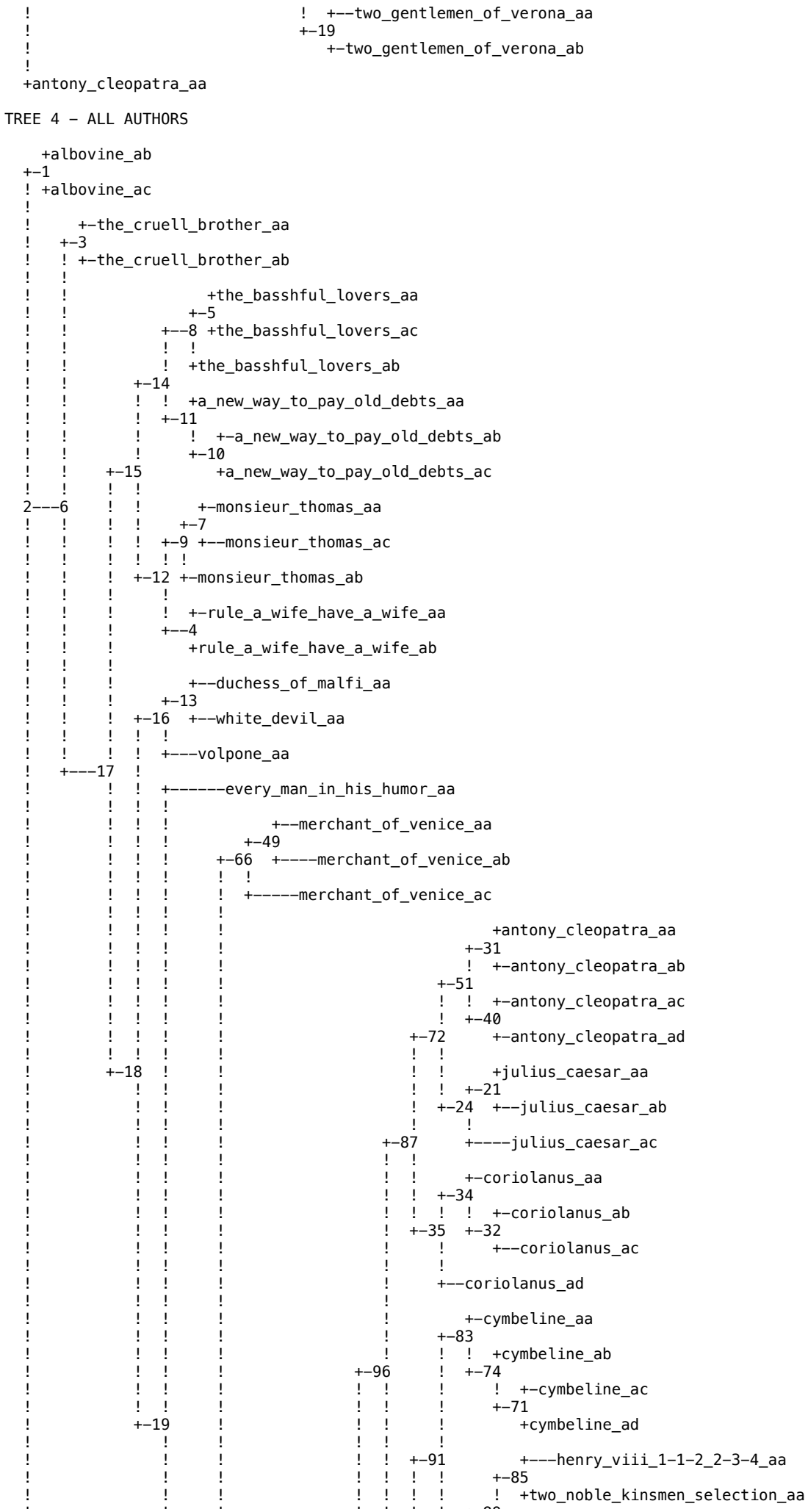

henry_v_ab
henry_v_ad
henry_v_ag
henry_v_ah
henry_v_ae
henry_v_af
henry_iv_pt1_aa
henry_iv_pt1_ab
henry_iv_pt1_ac
henry_iv_pt1_ad
henry_iv_pt1_af
henry_iv_pt1_ae
henry_iv_pt1_ag
henry_iv_pt2_af
henry_iv_pt2_aa
henry_iv_pt2_ab
henry_iv_pt2_ac
henry_iv_pt2_ad
henry_iv_pt2_ae
henry_iv_pt2_ah
troilus_and_cressida_aa
troilus_and_cressida_ad
troilus_and_cressida_af
troilus_and_cressida_ag
troilus_and_cressida_ah
troilus_and_cressida_ab
troilus_and_cressida_ac
troilus_and_cressida_ae
hamlet_aa
hamlet_ac
hamlet_ae
hamlet_ai
hamlet_ag
hamlet_ah
hamlet_ab
hamlet_af
hamlet_ad
othello_aa
othello_ab
othello_ac
othello_ad
othello_ae
othello_af



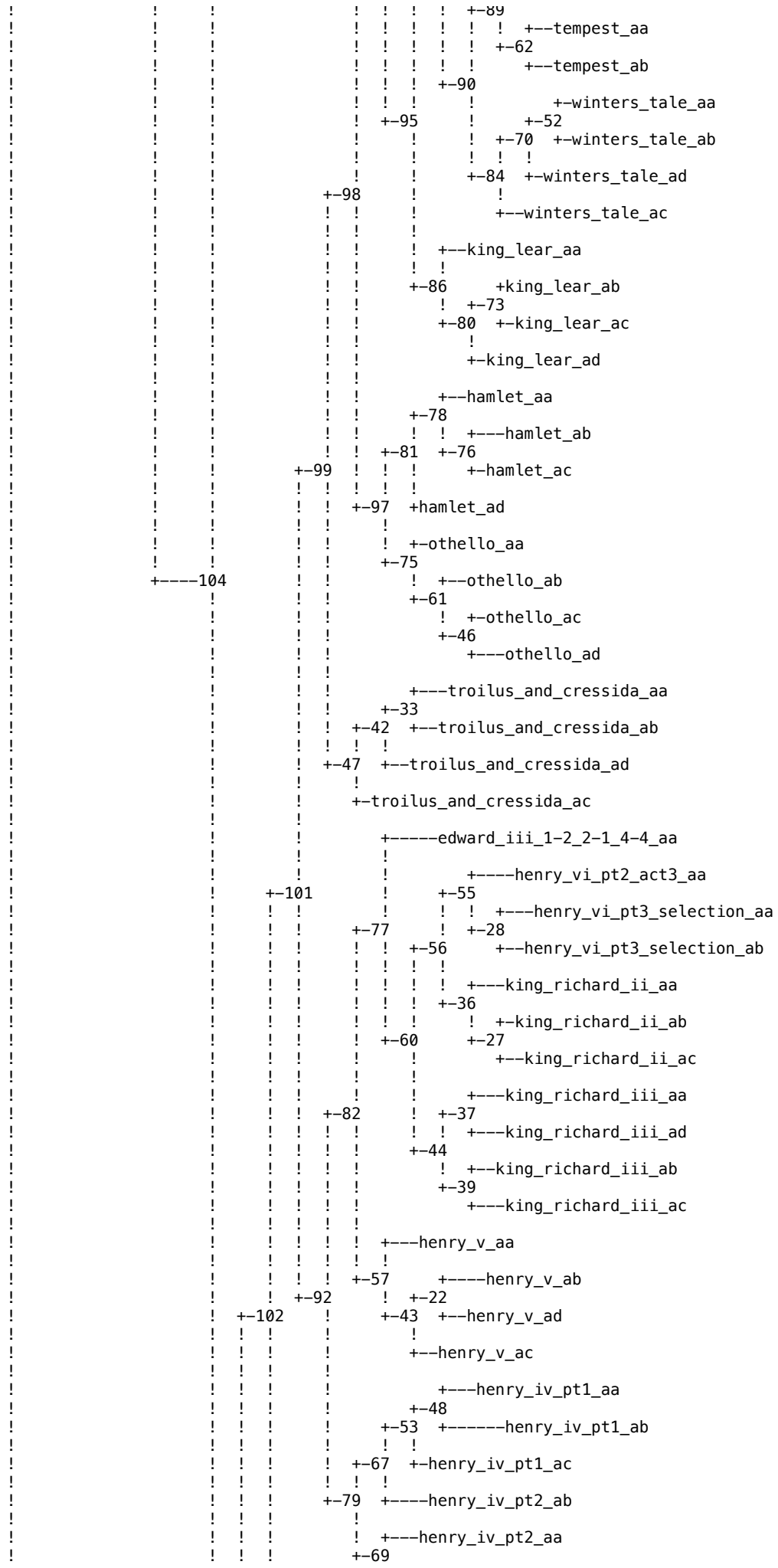
TREE 3 – SHAKESPEARE AND IBSEN (32KB)

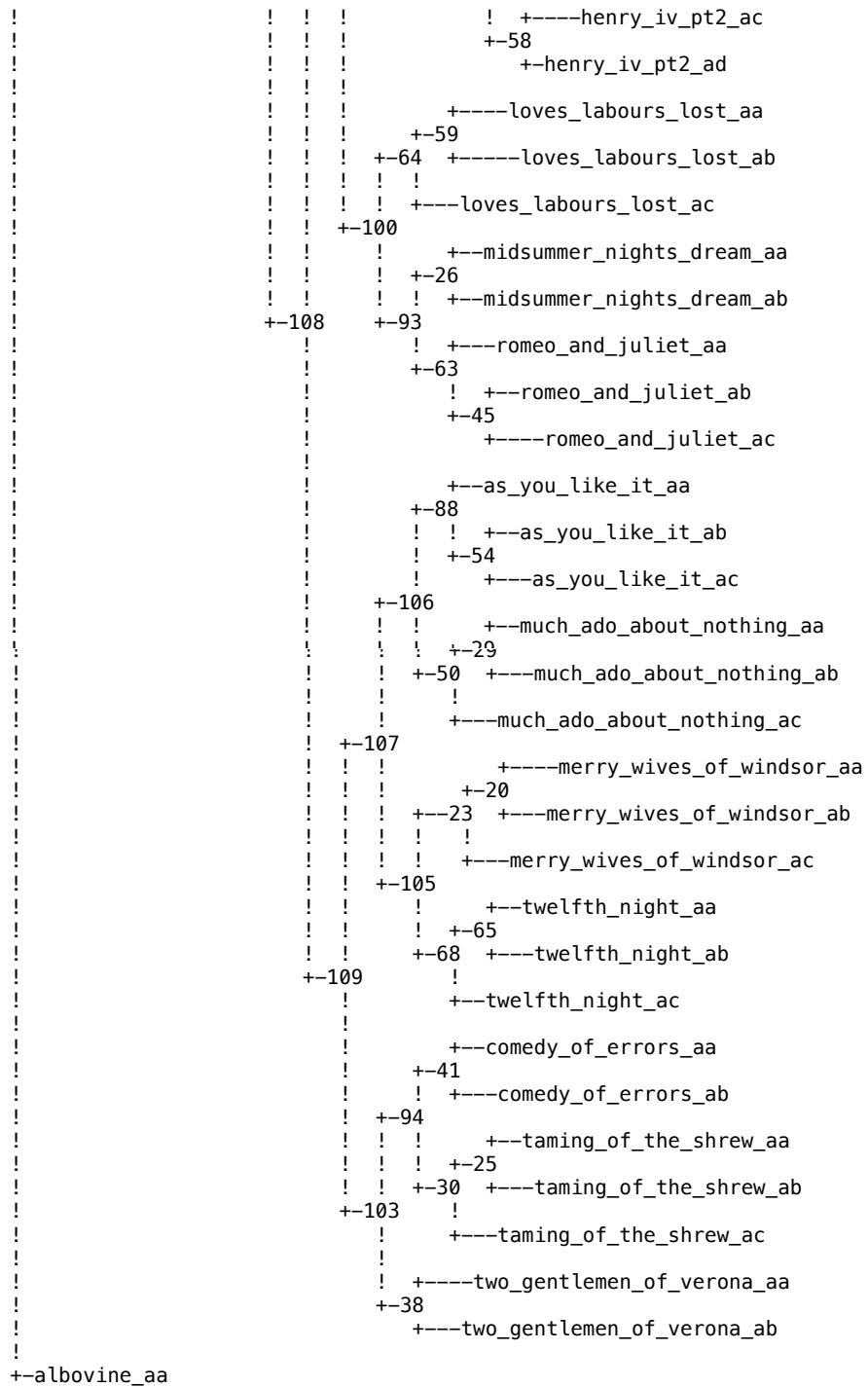

+julius_caesar_ac
+coriolanus_aa
14
36
+16
+coriolanus_ab
+23
+15
+coriolanus_ac
+coriolanus_ad
+cymbeline_aa
+65
+cymbeline_ab
+56
+61
+cymbeline_ac
+60
+cymbeline_ad
+78
+winters_tale_aa
+35
+54
+winters_tale_ab
+79
+66
+winters_tale_ad
+winters_tale_ac
+68
+tempest_aa
+44
+77
+tempest_ab
+80
+two_noble_kinsmen_selection_aa
+king_lear_aa
+63
+king_lear_ab
+69
+53
+king_lear_ac
+king_lear_ad
+hamlet_aa
+59
+hamlet_ab
+62
+58
+hamlet_ac
+82
+hamlet_ad
+81
+othello_aa
+57
+othello_ab
+39
+othello_ac
+22
+othello_ad
+troilus_and_cressida_aa
+17
+27
+troilus_and_cressida_ab
+32
+troilus_and_cressida_ad
+troilus_and_cressida_ac
+henry_viii_1-1-2_2-3-4_aa
+83
+67
+henry_v_aa
+42
+henry_v_ab
+6
+26
+henry_v_ad
+henry_v_ac
+henry_vi_pt2_act3_aa
+71
+37
+king_richard_ii_aa
+18
+king_richard_ii_ab
+38
+12
+king_richard_ii_ac
+henry_vi_pt3_selection_aa
+11
+40
+henry_vi_pt3_selection_ab


+-84
+-76
+-30
+-20
+-34
+-49
+-55
+-52
+-43
+-85
+-45
+-70
+-75
+-47
+-46
+-29
+-28
+-48
+-86
+-88
+-90
+-64
+-41
+-74
+-31
+-87
+-51
+-89
+-25
+-72
+-13
+-73
+-5
+-2
+-1
king_richard_iii_aa
21
king_richard_iii_ad
king_richard_iii_ab
king_richard_iii_ac
henry_iv_pt1_aa
24
henry_iv_pt1_ab
henry_iv_pt1_ac
henry_iv_pt2_ab
henry_iv_pt2_aa
52
henry_iv_pt2_ac
43
henry_iv_pt2_ad
dolls_house_aa
dolls_house_ab
dolls_house_ac
midsummer_nights_dream_aa
9
midsummer_nights_dream_ab
loves_labours_lost_aa
33
loves_labours_lost_ab
loves_labours_lost_ac
romeo_and_juliet_aa
46
romeo_and_juliet_ab
romeo_and_juliet_ac
merchant_of_venice_aa
28
merchant_of_venice_ab
merchant_of_venice_ac
as_you_like_it_aa
64
as_you_like_it_ab
41
as_you_like_it_ac
much_ado_about_nothing_aa
10
much_ado_about_nothing_ab
much_ado_about_nothing_ac
twelfth_night_aa
50
twelfth_night_ac
twelfth_night_ab
comedy_of_errors_aa
25
comedy_of_errors_ab
72
taming_of_the_shrew_aa
8
13
taming_of_the_shrew_ab
taming_of_the_shrew_ac
merry_wives_of_windsor_aa
4
5
merry_wives_of_windsor_ab
merry_wives_of_windsor_ac



TREE 4 – ALL AUTHORS


tempest_aa
tempest_ab
winters_tale_aa
winters_tale_ab
winters_tale_ad
winters_tale_ac
king_lear_aa
king_lear_ab
king_lear_ac
king_lear_ad
hamlet_aa
hamlet_ab
hamlet_ac
hamlet_ad
othello_aa
othello_ab
othello_ac
othello_ad
troilus_and_cressida_aa
troilus_and_cressida_ab
troilus_and_cressida_ad
troilus_and_cressida_ac
edward_iii_1-2_2-1_4-4_aa
henry_vi_pt2_act3_aa
henry_vi_pt3_selection_aa
henry_vi_pt3_selection_ab
king_richard_ii_aa
king_richard_ii_ab
king_richard_ii_ac
king_richard_iii_aa
king_richard_iii_ad
king_richard_iii_ab
king_richard_iii_ac
henry_v_aa
henry_v_ab
henry_v_ad
henry_v_ac
henry_iv_pt1_aa
henry_iv_pt1_ab
henry_iv_pt1_ac
henry_iv_pt2_ab
henry_iv_pt2_aa


henry_iv_pt2_ac
henry_iv_pt2_ad
loves_labours_lost_aa
loves_labours_lost_ab
loves_labours_lost_ac
midsummer_nights_dream_aa
midsummer_nights_dream_ab
romeo_and_juliet_aa
romeo_and_juliet_ab
romeo_and_juliet_ac
as_you_like_it_aa
as_you_like_it_ab
as_you_like_it_ac
much_ado_about_nothing_aa
much_ado_about_nothing_ab
much_ado_about_nothing_ac
merry_wives_of_windsor_aa
merry_wives_of_windsor_ab
merry_wives_of_windsor_ac
twelfth_night_aa
twelfth_night_ab
twelfth_night_ac
comedy_of_errors_aa
comedy_of_errors_ab
taming_of_the_shrew_aa
taming_of_the_shrew_ab
taming_of_the_shrew_ac
two_gentlemen_of_verona_aa
two_gentlemen_of_verona_ab
albovine_aa