Sex, death and sonnets
Musings of a software developer

Sigfrid Lundberg

slu@kb.dk
Digital Transformation
Royal Danish Library
Post box 2149
1016 Copenhagen K
Denmark

ABSTRACT

This note discusses how software can recognize sonnets, by analysis of text length, strophe structure and number of syllables per line. It also makes a simple content analysis based on word frequency analyses.

The results clearly shows that simple Unix™ for Poets analyses combines seamlessly with TEI markup and XML technologies.

Introduction

If there are any sonnets, do they rhyme and what are they about?

I have since many years been a great fan of the tutorial *Unix™ for Poets* by Kenneth Ward Church. This note is an investigation of what can be done with a corpus of literary text with very simple tools similar to the ones described by Church in his tutorial. I do not claim that there is anything novel or even significant in this text. Being a scientist, I think like a scientist and don't expect any deep literary theory here.

Finding poems

The ADL text corpus contains literary texts. Since the texts are encoded according to the TEI guidelines it is easy to find poetry in those files. Typically a piece of poetry is encoded as lines within line groups

A poem may look like this in the source. The poem is by Sophus Michaëlis (1883).

```
<div decls="#biblid68251">
   <head>Jeg elsker –</head>
   <lg>
      <l>Jeg elsker Himlens høje Harmoni,</l>
      <l>dens Purpurblomst, som blaaner i det Fjærne,</l>
      <l>den Fred, som risler ned fra Nattens Stjerne,</l>
      <l>det Glimt af Gud, der glider mig forbi;</l>
   </lg>
    <lg>
      <l>og Evighedens tavse Melodi,</l>
      <l>de svundne Slægters kaldende Orkester,</l>
      <l>et Tonehav om en usynlig Mester,</l>
      <l>en Klang af Gud, der bruser mig forbi;</l>
   </lg>
   <lg>
      <l>en magisk Magt fra Hjertets mørke Celle,</l>
      <l>de stærke Længsler, som mod Lyset vælde,</l>
      <l>Naturens evigunge Fantasi;</l>
   </lg>
   <lg>
      <l>det Liv, der spirer midt i selve Døden,</l>
      <l>den Sol, der stiger midt i Aftenrøden,</l>
      <l>– o Glimt af Gud, der glider mig forbi!</l>
   </lg>
   <p>
      <date>12. Septbr. 1893.</date>
   </p>
</div>
```

The default name space is declared as xmlns="http://www.tei-c.org/ns/1.0", which we in following refer to with the namespace prefix ’t’.

The poem comprises four line groups with four, four, three and three lines. That is a very common strophe structure (according to the Sonnets article in Wikipedia), at least in Scandinavia. It is not always like that, but they all contain 14 lines.

Shakespeare wrote often his 14 lines typographically in one strophe, whereas Francesco Petrarca wrote them in two strophes with eight and six lines, respectively (again see article Sonnets in Wikipedia).

To be more precise, a sonnet has one more characteristics than having 14 lines, the lines should be in iambic pentameter.

Finding sonnets

You can easily find all poems in the corpus based on a XPATH query like:

```
//t:div[t:lg and @decls]
```

We can use that query in XSLT like this:

```
<xsl:for-each select="//t:div[t:lg and @decls]">
   <xsl:if test="count(.//t:lg/t:l)=14">
      <!-- script's got to do what a script's got to do -->
   </xsl:if>
</xsl:for-each>
```

So we iterate over all ... s having line groups inside and have a '@decls' attribute containing a reference to metadata in the TEI header. The latter is not universal, but we use it in ADL and that attribute is only set on pieces that a cataloger has designated as a *work*. The decisions as to what is a work was based on the experience of what library patrons ask for at the information desk. I have implemented this using the shell script find_sonnet_candidates.sh and a transform sonnet_candidate.xsl

This transformation creates a long, sonnet_candidates.xml , table with data about the sonnet candidates it finds.

Approximately pentametric

Finding ... s having 14 lines of poetry isn't good enough. We are expecting iambic pentameter, don't we? To actually analyse the texts for their rythmical properties is beyond me, but we could make an approximation.

Iambic verse consists of feet with two syllables, i.e. if there are five feet per line we could say that iambic verse has approximately 10 vowels per line. It is an approximation since a iamb should have the stress on the second syllable (due to ignorance I ignore the musical aspect of this; we will include false positives since lines of poetry with five feet must not be **iambic.**

Any way, this script calculates the average number of vowels per line in poems with 14 lines:

```
<xsl:variable name="vowel_numbers" as="xs:integer *">
   <xsl:for-each select=".//t:lg/t:l">
      <xsl:variable name="vowels">
         <xsl:value-of select="replace(.,'[^iyeæøauoå]','')"/>
      </xsl:variable>
      <xsl:value-of select="string-length($vowels)"/>
   </xsl:for-each>
</xsl:variable>
<xsl:value-of select="format-number(sum($vowel_numbers) div 14, '#.####')"/>
```

We use the replace function and a regular expression to remove everything in each line except the vowels. Then we measure the string length which should equal the number of vowels per line and add them together for all lines in the poem. Finally we divide that sum with 14 and

get the average number of vowels per line.

For a sonnet it would be about 10, or occasionally a little more. Danish is a language rich in diftons, which could be another reason for lines deviating from the expected 10 vowels. In the Michaëlis poem quoted above it is 10.4.

Strophe structure

You can write a lot of nice poetry with 14 lines. Like Gustaf Munch-Petersen's en borgers livshymne with one strophe with one line, then three strophes with four lines and finally a single line. The number of syllables per line seem to decrease towards the end. Gustaf was a modernist. There are no fixed structures and very few rhymes i his poetry.

You can easily find out the strophe structure for each poem:

```
<xsl:variable name="lines_per_strophe" as="xs:integer *">
   <xsl:for-each select=".//t:lg[t:l]">
      <xsl:value-of select="count(t:l)"/>
   </xsl:for-each>
</xsl:variable>
<xsl:value-of select="$lines_per_strophe"/>
```

That is, iterate over the line groups in a poem, and count the lines in each of them.

I have summarized these data about all poems in ADL with 14lines. There are 243 of them (there might be more, but then they have erroneous markup).

You find these sonnet candidates in a table here sonnet_candidates.xml. Please, find an extract from it below.

sonnet candidates

File name (link to source)	Title (link to view)	Strophe structure	average number of vowels per line
		4 4 3 3	11.0
./aarestrup07val.xml	Jeg havde faaet Brev fra dig, Nanette		
		4 4 3 3	11.0714
./aarestrup07val.xml	Tag dette Kys, og tusind til, du Søde ...		
		4 4 3 3	11.5
./aarestrup07val.xml	Sonet		
		14	8.7143
./brorson03grval.xml	1.		
		14	10.8571
./claussen07val.xml	SKUMRING		

		4 4 3 3	13.8571
./claussen07val.xml	MAANENS TUNGSIND		
		14	6.7143
./jacobjp08val.xml	I Seraillets Have		

Sophus Claussen’s first poem may or may not be a sonnet, Brorson’s poem is not. All of those with strophe structure 4 4 3 3 are definitely sonnets, as implied by strophe structure and the "approximately pentametric" number of vowels per line (and, by the way, Aarestrup often points out that he is actually writing sonnets in text or titles).

Then we have the rhymes

Beauty is in the eye of the beholder, says Shakespeare. I believe that he is right. Then, however, I would like to add that the rhymes and meters of poetry (like the pentameter) is in the ear of listener. It is time consuming to read houndreds of poems aloud and figure out the rhyme structure. So an approximate idea of the rhymes could be have comparing the verse line endings.

This is error prone, though. Consider this sonnet by P.M. Møller

> SONET
>
> Den Svend, som Tabet af sin elskte frister,
> Vildfremmed vanker om blandt Jordens Hytter;
> Med Haab han efter Kirkeklokken lytter,
> Som lover ham igen, hvad her han mister.
>
> Men næppe han med en usalig bytter,
> Hvis Hjerte, stedse koldt for Elskov, brister,
> Som sig uelsket gennem Livet lister,
> Hans Armod kun mod Tabet ham beskytter.
>
> Til Livets Gaade rent han savner Nøglen,
> Hver Livets Blomst i Hjærtets Vinter fryser,
> Han gaar omkring med underlige Fagter.
>
> Ræd, Spøgelser han ser, naar Solen lyser,
> Modløs og syg, foragtet han foragter
> Det skønne Liv som tom og ussel Gøglen.

The the last syllable of the eight first lines are the same ’-ter’. If you use some script to compare the endings you’ll only find single syllable rhymes and miss double syllable ones rhymes. I.e., you can erroneously categorize feminine rhymes (with two syllables) as masculine ones (with one syllable). (Sorry, I don’t know a politically correct vocabulary for these concepts.)

In order to understand what we hear when reading, we have to consider ’-ister’ and ’-ytter’. I.e., it starts with rhyme structure ’abbabaab’ not ’aaaaaaaa’. Furthermore, it continues ’cdedec’.

I have written a set of scripts that traverse the sonnet_candidates.xml table. Transform that file using iterate_the_rhyming.xsl selects poems with 14 lines and strophe structure 4 4 3 3. It

generates a shell script which when executed pipes the content through other scripts that retrieve content, remove punctuation and finally detags them. The actual text is then piped through a perl script that analyse the endings according to the silly and flawed method described above.

It works, sort of, until it doesn't. For poems with 4 4 3 3 strophe structure, you can find the result in rhymes_3chars.text and rhymes_2chars.text for three and two letter rhymes, respectively. Run

```
grep -P '^[a-q]{14}' rhymes_3chars.text   | sort | uniq -c | sort -rn
```

to get a list of rhyme structure and their frequencies. The rhyme structures that occur more than twice are:

```
6 abbaabbacdecde
5 abbaabbacdcdcd
4 abcaadeafgghii
4 abbaabbacdcede
3 abcaadeafghgig
```

This silly algorithm does actually give two of the most common rhyme structure for sonnets, but misses a lot of order in the remaining chaos:

`abbaabbacdcdcd`

and

`abbaabbacdecde`

So while it may fail more often than it succeeds, the successes give results that are reasonable.

The rhyme structure abbaabbacdecde is one is the most common ones found. Also it is one of the socalled Petrarchan rhyme schemes (Eberhart, 2018).

What are the sonnets about?

Any piece of art is meant to be consumed by humans. Poems should ideally be understood when read aloud and listened to. By humans.

The cliché says that art and literature is about what it means to be human. Could we therefore hypothesize that the sonnets address this from the point of view of dead Danish male poets who wrote sonnets some 100 – 200 years ago?

Assume that, at least as a first approximation, the words chosen by poets mirror those subjects. For instance, if being human implies lethality, we could, on a statistical level hypothesize that words like "mourning", "grief", "death", "grave", etc appear in the sonnet corpus more than in a random sample of text. The opposites would also be expected: Concepts related to "love", "birth", "compassion" belong to the sphere of being human.

I have detagged the poems with 14 lines and strophe structure 4 4 3 3, tokenized their texts and calculated the word frequencies. As a matter of fact, I've done that in two ways:

(i) The first being doing a classical tokenization followed by piping the stuff through

```
sort | uniq -c | sort -n
```

such that I get a list of the 4781 Danish words that are used in our sonnet sample, sorted by their frequencies.

(ii) The second way is the same, but I do it twice, once for each sonnet such that I get a list of words for each sonnet. Then I repeat that for the concatenated lists for all sonnets.

This means that I get

- one list of word frequencies in the entire sample and
- a second list giving not of the number of occurences of each word, but the number of sonnets the word occurs in.

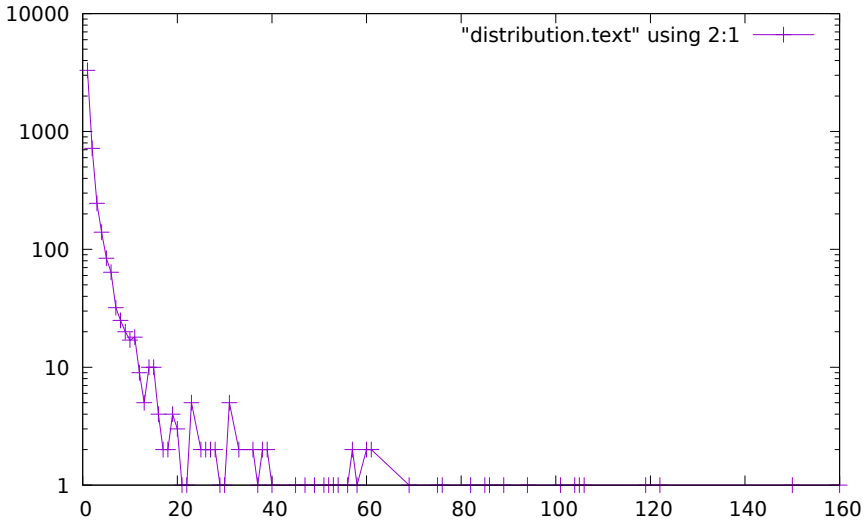
There are 160 sonnets in the selection, and the most frequent word occurs in all of them. These are the fifteen most commont word measured by the number of sonnets they occur in

```
75 du
76 sig
82 er
85 jeg
86 det
89 for
94 den
101 paa
104 en
105 af
106 til
119 som
122 med
150 i
160 og
```

and this is the list of the same thing, but measured as the grand total occurrence of the words in the corpus

```
109 min
130 for
144 du
148 er
155 paa
164 til
167 det
169 den
173 af
206 en
217 med
229 som
246 jeg
382 i
588 og
```

As you can see this corroborates the established observation that the most frequent words in a corpus hardly ever describes the subject matter of texts (the words are conjunctions, pronouns, prepositions and the like). The distribution of the number of sonnets the words appear in:



The distribution shows number of words graphed against number of sonnets. There are 3304 words occurring in just one sonnet. The leftmost, and highest, point on the graph has the coordinate (1,3304).

There is just one word appearing in all 160 sonnets. It is ’og’ meaning ’and’ correspoding to the rightmost point on the graph which has the coordinate (160,1). As a rule of thumb the most common words are all conjunctions, next to them comes prepositions and after those come pronomina.

The distribution.text is generated from poem_frequencies.text using (the line has been folded)

```
sed 's/ [a-z]*$//' poem_frequencies.text | sort | uniq -c |
sort -n -k 2 > distribution.text
```

See above. Column 1 is plotted against column 2.

In this particular corpus, it seems that **aboutishness** start at words occuring in about 25% of the sonnets, or less. I.e., words occuring in 40 sonnets, or fewer.

In what follows, I have simply used the utility `grep` find words and derivates in the file poem_frequencies.text mentioned above.

As example we have death, dead and lethal etc (basically words containing *død*) in a number of sonnets. In the left column the number of sonnets containing the word. These appear in about 7% of the sonnets.

```
1 dødehavet
1 dødeklokker
1 dødelige
1 dødeliges
1 dødningvuggeqvad
1 dødsberedthed
1 glemselsdøden
1 udødeliges
2 dødes
5 dødens
9 død
9 døden
11 døde
```

There are interesting derivatives and compound words on the list. Like *dødsberedthed* meaning preparedness for death. *Glemselsdøden* refers, I belie ve, to the death or disappearance due to the disappearance of traces or memories of someone who belonged to generations.

Love (elskov) is not as popular as death (about 5% of the sonnets).

```
1 elskoven
1 elskovsbrev
1 elskovsbrevet
2 elskovsild
6 elskovs
7 elskov
```

elskovsild means the fire of love. *elskovsbrev* has to be love letter. *women (kvinde)* are not as popular as love

```
1 dobbeltkvinde
1 kvindens
1 kvindetække
4 kvinder
```

Men more than women, and in particular words implying bravery and male virtues

```
1 baadsmandstrille
1 dobbeltmand
1 ejermand
1 manddom
1 manddomstrods
1 manden
2 mand
2 manddoms
5 mandens
```

Remember that these sonnets are by men. *mandom* implies a man' s existence as a grownup man. Originally, in old norse , mand meant, just as in Old English, human. That, however, was when it was doubtful if women were actually human. Baadsmandstrille is a derivative of baadsmand (boatswain) which is another name for a sailor or petty officer. A baadsmandstrille is presumably a song sung by sailors.

Graves occur, for some reason, less than deaths

```
1 begravet
1 graven
1 gravene
1 gravhøi
1 indgraves
3 grav
3 grave
4 gravens
```

indgraves is most likely a kind of *homonym* , if you look up that sonnet it is clear that it means engrave. There both the verb in past tense begravet (buried) from begrave (as in bury) and grav (as in grave) and gravhøi (tumulus).

Conclusions

I think I could go on studying this for quite some time. However, I have to conclude this here, before the actual conclusions. There are interesting things to find here, though. Some of them are possible to study using simple methods, such as those described by Kenneth Ward Church in his *Unix™ for Poets* .

The preliminary result from my armchair text processing exercise supports the notion that life was already in early modern Europe about sex, death and rock n'roll. Since rock wasn't there just yet, people had to be content with sonnets for the time being.

References

Church, Kenneth Ward, [date unknown]. *Unix™ for Poets*
https://web.stanford.edu/class/cs124/kwc-unix-for-poets.pdf

Det Kgl. Bibliotek, and Det Danske Sprog- og Litteraturselskab, 2000 - 2022. *The ADL text corpus*
https://github.com/kb-dk/public-adl-text-sources

Eberhart, Larry, 2018. Italian or Petrarchan Sonnet. In: *Every Sonnet: The sonnet forms database*
https://poetscollective.org/everysonnet/tag/abbaabbacdecde/#post-119

Hendecasyllable. In:*W ikipedia*
https://en.wikipedia.org/wiki/Hendecasyllable

Iambic pentameter. In:*W ikipedia*
https://en.wikipedia.org/wiki/Iambic_pentameter

Michaëlis, Sophus, 1883. Jeg elsker —. In: *Solblomster*
https://tekster.kb.dk/text/adl-texts-michs_03-shoot-workid68251

Old Norse. In: *Wikipedia*
https://en.wikipedia.org/wiki/Old_Norse

Sonnet. In:*W ikipedia*
https://en.wikipedia.org/wiki/Sonnet

The TEI Consortium, 2022. *TEI P5: Guidelines for Electronic Text Encoding and Interchange*
https://tei-c.org/release/doc/tei-p5-doc/en/html/index.html

The TEI Consortium, 2022. Passages of Verse or Drama. In: *TEI P5: Guidelines for Electronic Text Encoding and Interchange*
https://tei-c.org/release/doc/tei-p5-doc/en/html/CO.html#CODV