

Transcriptional Implicature

Michael Sperberg-McQueen
Claus Huitfeldt
Yves Marcoux

Balisage 2026

Michael Sperberg-McQueen



© B. Tommie Usdin

Transcription

As in textual scholarship and critical editing

(Not as in music, linguistics, genetics...)

Everybody knows what transcription is --

-- but nobody does it the same way.

- Preserve line breaks?
- Expand abbreviations?
- Include deletions?
- Correct slips of the pen?

Confusion and acrimony...

Transcription

Confusion and acrimony...

Vander Meulen and Tanselle (1999) regret that:

“numerous discussions of the subject, including some of the most influential, allow for making certain classes of alteration when transcribing manuscript texts, **as if a conscious program of alteration is compatible with the concept of transcription**” (pp. 201–2).

"Statement of practice" of a project

- Rules governing how transcription is performed
- Hard to express in a precise yet readable way
- Some rules "go without saying"
- Thus, difficult to know exactly how *similar* the exemplar and transcript are expected to be

Our endeavour

- Provide a formal framework in which various flavours of similarity can be precisely defined
 - Similarity of two documents *is a theorem!*
- Propose **Transcriptional Implicature** as capturing the rules that "go without saying" in a project
- Propose **Default Transcriptional Implicature**, relative to which the rules of any project can be defined
- Working environment in which the similarity of two documents can be verified by an automatic theorem prover

Caveat 1

- For t to be a transcript of e:
 - e must antedate t
 - t must be made in the presence of e
 - t must be made with a certain intention regarding e
 - t must be "similar" in some way to e...
- We limit ourselves to describing the *similarity relation* between e and t

Caveat 2

- We specifically *do not aim* at:
 - Describing the perceptual / cognitive / psychological processes at play in the practice of transcription
 - Being *prescriptive* about how transcription should be done
 - ~~Do's and don'ts~~
 - ~~Best practices~~
 - ~~etc.~~

Sketch of formal framework

- First-order predicate logic
- Exemplar (e) and transcript (t) represented as *constants*
 - One constant for each *token* and *type*
- Axioms describing e , t , and how they map
- Small number of general axioms (well-formedness, OWA)
- Axioms defining T-similarity of two documents
- $\text{T-similar}(e, t)$ is a theorem $\leftrightarrow$ t is the result of applying to e the transcription rules in the statement of practice

Type/token analysis

- mark discernible physical arrangement in document
- token mark significant for the "text" of document
- type character, letter, word, sentence, ...
- Each token is mapped to a single type
- Tokens / types can be *compound*
 - Various topologies: sequence, set, bag, ...
 - Document as a whole is (usually) a compound token
 - In our examples: sequence only
- Type systems and type repertoires
- Bidirectional mapping between tokens of e and of t

Toy example

e: *Elizabeth went to Essex*

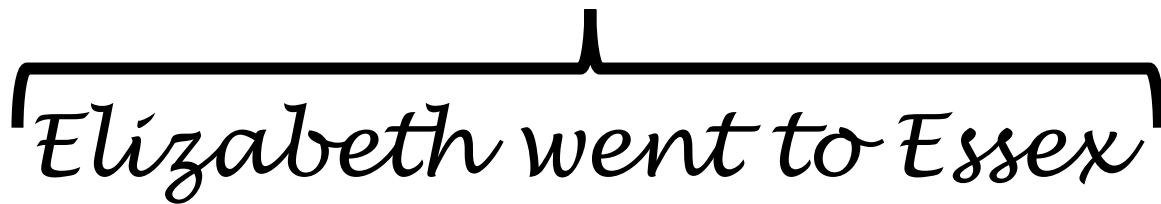
t: Elizabeth went to Essex

- "Empty" statement of practice

Toy example - exemplar

compound token:

e



Elizabeth went to Essex

basic tokens:

e1

e2

e3

e4

Formally:

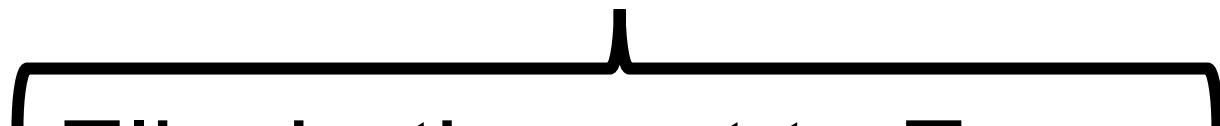
```
typeof(e1, t_Elizabeth) &  
typeof(e2, t_went) &  
typeof(e3, t_to) &  
typeof(e4, t_Essex) &
```

```
typeof(e, t_Elizabeth_went_to_Essex)
```

Toy example - transcript

compound token:

t



Elizabeth went to Essex

basic tokens:

t1

t2

t3

t4

Formally:

```
typeof(t1, t_Elizabeth) &  
typeof(t2, t_went) &  
typeof(t3, t_to) &  
typeof(t4, t_Essex) &
```

```
typeof(t, t_Elizabeth_went_to_Essex)
```

Recall:

```
typeof(e1, t_Elizabeth) &  
typeof(e2, t_went) &  
typeof(e3, t_to) &  
typeof(e4, t_Essex) &
```

```
typeof(e, t_Elizabeth_went_to_Essex)
```

Mapping with e:

```
transcript(e1,t1) &  
transcript(e2,t2) &  
transcript(e3,t3) &  
transcript(e4,t4) &
```

```
transcript(e,t)
```

T-similarity of default t.i. - first try

- All pairs of tokens mapped between e and t are of the same type

`typeof(t1) = typeof(e1)`

`typeof(t2) = typeof(e2)`

`typeof(t3) = typeof(e3)`

`typeof(t4) = typeof(e4)`

`typeof(t) = typeof(e)`

- *Too restrictive:* cannot accommodate any relaxation of the transcription rules

It must be possible to define
T-similarity so that it is satisfied by:

e: *Elizabeth went ~~with~~^{to} Essex.*

t: *Elizabeth went to [the Earl of] Essex*

What is needed...

- Special status for tokens that do not correspond to any token on the "other side" (exemplar or transcript)
 - In example: ~~with~~ and *[the Earl of]*
- Possibility of declaring types as "similar", though not identical
 - In example: *Essex* and Essex

Elizabeth went ~~with~~ ^{to} Essex.

e1

e2

e3

e4

e5

Elizabeth went to [the Earl of] Essex

t1

t2

t3

t4

t5

t6

t7

N.B.: For simplicity, from now on, we omit compound tokens

Formally

Documents:

```
typeof(e1, t_Elizabeth) &  
typeof(e2, t_went) &  
typeof(e3, t_with) &  
typeof(e4, t_to) &  
typeof(e5, t_Essex)
```

```
typeof(t1, t_Elizabeth) &  
typeof(t2, t_went) &  
typeof(t3, t_to) &  
typeof(t4, t_the) &  
typeof(t5, t_Earl) &  
typeof(t6, t_of) &  
typeof(t7, t_Essex)
```

Mapping between e and t:

```
transcript(e1,t1) &  
transcript(e2,t2) &  
transcript(e4,t3) &  
transcript(e5,t7)
```

Type similarity:

```
typesimilar(t_Essex, t_Essex)
```

T-similarity of default t.i. - revised

- All pairs of *normal* tokens mapped between e and t are of the *similar* (or equal) types

`typeof(t1) ≈ typeof(e1)`

`typeof(t2) ≈ typeof(e2)`

`typeof(t3) ≈ typeof(e4)`

`typeof(t7) ≈ typeof(e5)`

Added flexibility

- If all tokens are normal and no unequal types are considered similar
 - Equivalent to original formulation
- Turns out most cases can be handled with special tokens and pairs of *similar* types
- This is our proposed formulation of T-similarity for the *default transcriptional implicature*

T-similarity as conjuncts

- For convenience in discussing cases that do *not* achieve T-similarity, its definition can be broken down into five components:

T-similarity $\leftrightarrow$

reciprocity & single_valued_mappings &
completeness & purity & type_similarity

e: *Elizabeth went ~~with~~^{to} Essex.*

t: Elizabeth went to *[the Earl of]* Essex

How formalize two (non-digital) textual objects and the relation between them?

e: *Elizabeth went ~~with~~^{to} Essex.*

t: *Elizabeth went to [the Earl of] Essex*

Proving that e and t are T-similar:

1. XML representation of e
2. XML representation of t
3. Transform 1 and 2 to a FOL representation of both*
4. Prove that 3 implies T-similarity

e: *Elizabeth went ~~with~~^{to} Essex.*

t: Elizabeth went to *[the Earl of]* Essex

XML representation of e:

```
<p> Elizabeth went <del>with</del> <add>to</add> Essex</p>
```

XML representation of t:

```
<p> Elizabeth went to <supplied>the Earl of</supplied>  
<choice><orig>Essex</orig><reg>Essex</reg></choice></p>
```

FOL representation of e and t

General axioms and definitions (excerpt)

fof(transcript_domain, axiom,
! [X,Y] : (transcript(X,Y) => (e_token(X) & t_token(Y)))).

fof(typeof_domain, axiom,
! [X,Y] : (typeof(X,Y) =>
((t_token(X) | e_token(X) | st_token(X) | se_token(X)) & type(Y)))).

fof(exemplar_and_transcript_inverse, axiom,
! [X,Y] : (exemplar(X,Y) <=> transcript(Y,X))).

fof(at_most_one_type_per_token, axiom,
! [X,Y,Z] : ((typeof(X,Y) & typeof(X,Z)) => Y=Z)).

... ..

...

Case-specific facts

fof(case_specific_facts, axiom,

typeof(e1, t_Elizabeth) &
typeof(e2, t_went) &
typeof(e3, t_with) &
typeof(e4, t_to) &
typeof(e5, t_Essex) &

typeof(t1, t_Elizabeth) &
typeof(t2, t_went) &
typeof(t3, t_to) &
typeof(t4, t_the) &
typeof(t5, t_Earl) &
typeof(t6, t_of) &
typeof(t7, t_Essex) &

typesimilar(t_Essex, t_Essex) &

transcript(e1,t1) &
transcript(e2,t2) &
transcript(e4,t3) &
transcript(e5,t7) &

... ..
... ..
...

Conjecture

fof(t_similar, conjecture,
reciprocity & single_valued_mappings &
completeness & purity & type_similarity).

Conjecture

fof(t_similar, conjecture,
reciprocity & single_valued_mappings &
completeness & purity & type_similarity).

Vampire says:

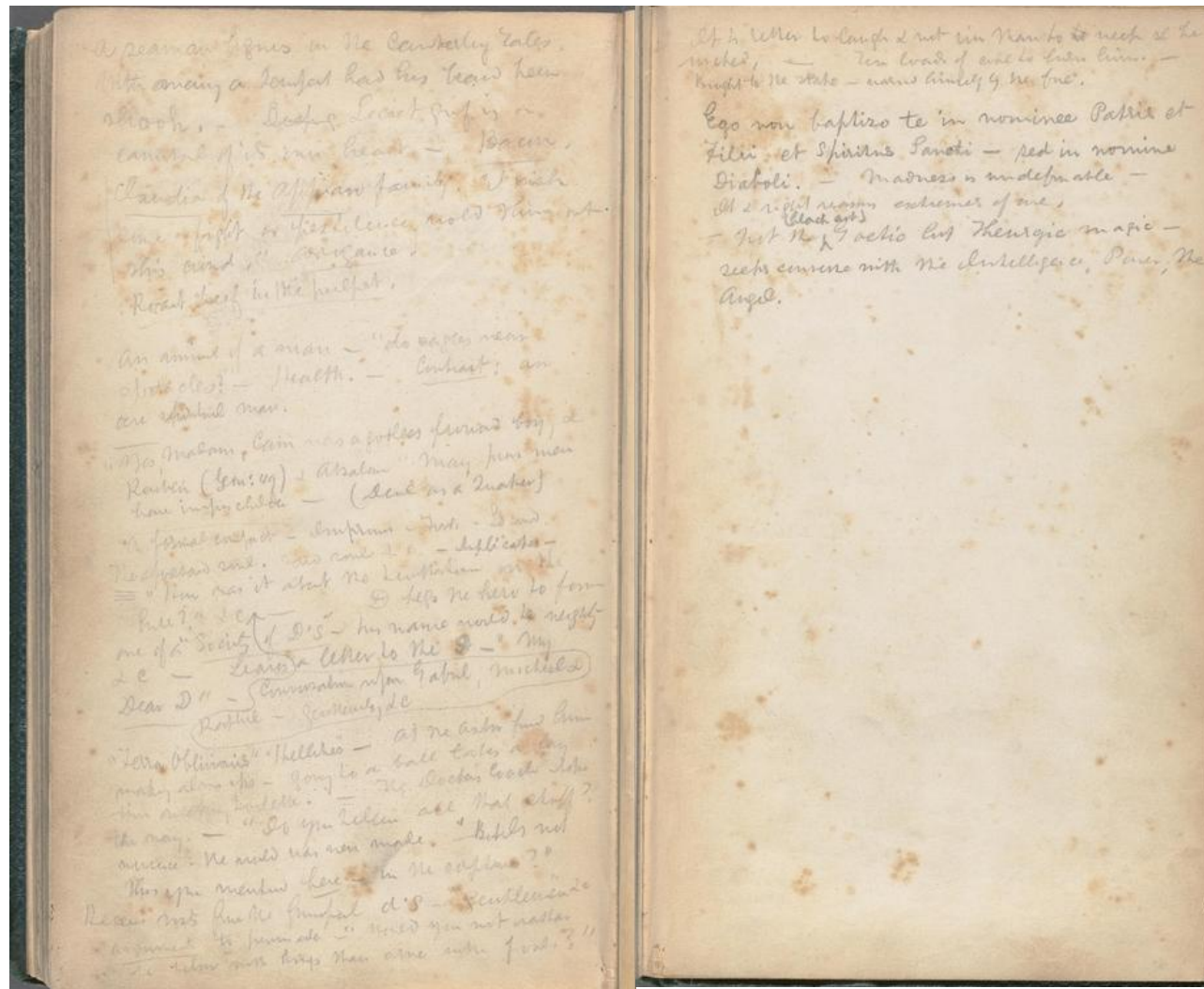
% Refutation found.
% SZS **status Theorem** for Exceptional
% -----
% Version: Vampire 4.9 (commit)
% Termination reason: Refutation

% Memory used [KB]: 1050
% Time elapsed: 0.046 s

Melville's notes in a Shakespeare volume

(Northwestern-Newberry edition.
Hayford, Harrison, Parker, Tanselle 1998)

Exemplar



2026-08-04

32

[on verso of last leaf of Volume VII, page [524]]

1 A seaman figures in The Canterbury Tales.
2 With <a>many a tempest had his beard been
3 shook. — <S><Deep g> Secret grief is a
4 cannibal of its own heart — Bacon.
5 | Claudia of the Appian family, "I wish
6 | some fight or pestilence would thin out
7 | this crowd." Arrogance.
8 Roast beef in the pulpet.

9 An animal of a man — "do eagles wear
10 spectacles?" — Health. — Contrast: an
11 over spiritual man.

12 "Yes, Madam, Cain was a godless froward boy, &
13 Reuben (Gen:49) & Absalom" Many pious men
14 have impious children — (Devil as a Quaker)

15 A formal compact — Imprimis — First — Second.
16 The aforesaid soul. said soul &c — Duplicates —
17 ≡ "How was it about the temptation on the
18 hill?" &c [inserted later below in lines 21–21b after Dear D" — and circled
with guideline to caret here Conversation upon Gabriel, Micheal & /
Raphel — gentlemanly &c] — D begs the hero to form
19 one of a "Society of D's" — his name would be weighty
20 &c — Leaves a letter to the D — "My
21 Dear D" — [later insertion in lines 21–21b, reported in line 18]

22 "Terra Oblivionis" "Hellites" — At the Astor find him
23 ?making ?almxxxx — going to a ball takes a long
24 time making toilette. — The Doctor's coach stops
25 the way. — "Do you beleive all that stuff?
26 nonsense — the world was never made. — [add "But] Is not
27 this you mentioned here — in the scriptures?"
28 Receives visits from the principal d's — "Gentlemen" &c.
29 Arguments to persuade — "Would you not rather
30 be below with kings than above with fools?"

[on recto of last blank leaf of Volume VII, page [523]].

31 It is better to laugh & not sin than to <be> weep & be
32 wicked. — Ten loads of coal to burn him. —
33 Brought to the stake — warmed himself by the fire.

34 Ego non baptizo te in nomine Patris et
35 Filii et Spiritus Sancti — sed in nomine
36 Diaboli. — Madness is undefinable —
37 It & right reasons extremes of one.

38 —Not the [inserted later above line with caret below (black art)] Goetic
but Theurgic magic —

39 seeks converse with the Intelligence, Power, the
40 Angel.

Statement of practice (p. 967)

[...]	revision or insertion enclosed in square brackets was made later than initial inscription of leaf
< ... >	letters or words enclosed in diamond brackets were canceled by lining out
< ... >word	letter(s) or word(s) written over are enclosed in diamond brackets closed up to the following word or letter that was superimposed
?word	prefixed question mark indicates conjectural reading
xxxx	undeciphered letters (number of x's approximates numbers of letters involved)
	all words in roman are Melville's
	all words in italics <i>outside brackets</i> are words Melville underlined
	all words in italics <i>inside brackets</i> are editorial

Exemplar XML (excerpt)

```
<text>
  <body>
    <p>
      <lb/>A seaman figures in The Canterbury Tales.
      <lb/>With <del>a</del>many a tempest had his beard been
      <lb/>shook. - <del>S</del><del>Deep g</del> Secret grief is a
      <lb/>cannibal of its own heart - <emph>Bacon</emph>.
      <lb/>| <emph>Claudia</emph> of the <emph>Appian</emph> family, "I wish
      <lb/>| some fight or pestilence would thin out
      <lb/>| this crowd." <emph>Arrogance.</emph>
      <lb/><emph>Roast beef in the pulpet.</emph>
    </p>
    <p>
      <lb/>An animal of a man - "do eagles wear
```

Transcript XML (excerpt)

```
<text>
  <front>
    <docTitle>
      <titlePart>NOTES IN A SHAKESPEARE VOLUME    <span>969</span></titlePart>
    </docTitle>
  </front>
  <body>
    <fw>[on verso of last leaf of Volume VII, page [524]]</fw>
    <p>
      <lb n="1"/>A seaman figures in The Canterbury Tales.
      <lb n="2"/>With <del>a</del>many a tempest had his beard been
      <lb n="3"/>shook. - <del>S</del> <del>Deep g</del> Secret grief is a
      <lb n="4"/>cannibal of its own heart - <emph>Bacon</emph>.
      <lb n="5"/><metamark>|</metamark> <emph>Claudia</emph> of the
        <emph>Appian</emph> family, "I wish
      <lb n="6"/><metamark>|</metamark> some fight or pestilence would thin out
      <lb n="7"/><metamark>|</metamark> this crowd." <emph>Arrogance.</emph>
      <lb n="8"/><emph>Roast beef in the pulpet.</emph>
    </p>
    <p>
      <lb n="9"/>An animal of a man - "do eagles wear
```

FOL representation

```
fof(case_specific_facts, axiom,  
    typeof(e1, t_A) &  
    typeof(e2, t_seaman) &  
    typeof(e3, t_figures) &  
    typeof(e4, t_in) &  
    ...  
    typeof(e287, t_Intelligence) &  
    typeof(e288, t_Power) &  
    typeof(e289, t_the) &  
    typeof(e290, t_Angel) &  
  
    typeof(t1, t_NOTES) &  
    typeof(t2, t_IN) &  
    typeof(t3, t_A) &  
    typeof(t4, t_SHAKESPEARE) &  
    ...  
    typeof(t348, t_Intelligence) &  
    typeof(t349, t_Power) &  
    typeof(t350, t_the) &  
    typeof(t351, t_Angel) &
```

```
typesimilar(t_almanacks, t_almxxxx) &
```

```
...  
    transcript(e1,t17) &  
    transcript(e2,t18) &  
    transcript(e3,t19) &  
    transcript(e4,t20) &  
  
...  
    transcript(e287,t348) &  
    transcript(e288,t349) &  
    transcript(e289,t350) &  
    transcript(e290,t351) &
```

```
... ..
```

```
... ..
```

```
...
```

Conjecture

fof(t_similar, conjecture,
reciprocity & single_valued_mappings &
completeness & purity & type_similarity).

Conjecture

fof(t_similar, conjecture,
reciprocity & single_valued_mappings &
completeness & purity & type_similarity).

Vampire says:

```
% Refutation found.  
% SZS status Theorem for Exceptional  
% -----  
% Version: Vampire 4.9 (commit )  
% Termination reason: Refutation  
  
% Memory used [KB]: 4453111  
% Time elapsed: 3083.273 s
```

Closing remarks

Proof of concept, i.e. of the applicability and usefulness of the notion of transcriptional implicature.

- It is actually possible to automate a process going from two XML(TEI)-encoded documents to a formal proof of whether they are T-similar or not.
- Obviously not a production tool: The Melville example is < 300 words, thousands of FOL statements, takes minutes to process
- Still, we are impressed by Vampire. (There may of course be other and even better tools.)

Closing remarks

In 2008, we proposed two models of transcription:

The grapheme sequence model.

The readings model.

- We found the grapheme sequence model unsatisfactory

In 2010, we proposed an extension of the type-token distinction

In this paper:

- We have only applied the grapheme* sequence model
- We have not fully exploited the 2010 extensions of the type-token distinction

Closing remarks

On the other hand:

The notion of type similarity, and the fact that the representation of the exemplar must reflect that of the transcript,

- (Special exemplar tokens and special transcript tokens)

may inspire further work developing this model to a readings model.

Furthermore, it would be interesting to investigate whether the methods used here might inspire useful ways of comparing other kinds of relations between documents than that between exemplars and their transcripts.

Thank you

