Growing Trees:
Non-Linear Incremental Parsing during Writing

Cerstin Mahlow
Institut für Deutsche Sprache
mahlow@ids-mannheim.de

Fred-Jelinek Seminar, Prague, April 4, 2016

Incremental Parsing

- Computer Science
  - ...
  - ...
  - ...
- NLP
  - ...
  - ...
  - ...

Incremental Parsing

- Computer Science
  - on-going process during coding
  - parsing of the structure so far
  - update the parse according to changes
- NLP
  - batch process
  - processing of text
  - parse a sentence word-by-word

Incremental Parsing

- Computer Science
  - on-going process during coding
  - parsing of the structure so far
  - update the parse according to changes
- NLP
  - batch process
  - processing of text
  - parse a sentence word-by-word

Incremental Parsing

- Computer Science
  - on-going process during coding
  - parsing of the structure so far
  - update the parse according to changes
- NLP
  - batch process
  - processing of text
  - parse a sentence word-by-word

Incremental Parsing

- Computer Science
  - on-going process during coding
  - parsing of the structure so far
  - update the parse according to changes
- NLP
  - batch process
  - processing of text
  - parse a sentence word-by-word

Incremental Parsing

- Computer Science
  - on-going process during coding
  - parsing of the structure so far
  - update the parse according to changes
- NLP
  - batch process
  - processing of text
  - parse a sentence word-by-word

Writing

- non-linear creation of text
- differs from:
  - speaking
  - hearing
  - reading

Writing

- non-linear creation of text
- differs from:
  - speaking
  - hearing
  - reading

Writing

- non-linear creation of text
- differs from:
  - speaking
  - hearing
  - reading

Writing

- non-linear creation of text
- differs from:
  - speaking
  - hearing
  - reading

Writing

- non-linear creation of text
- differs from:
  - speaking
  - hearing
  - reading

Challenges in Live Processing of the Writing Process

- non-linear
- revisions
- big data
- media discontinuity
- author-dependent

Challenges in Live Processing of the Writing Process

- non-linear
- revisions
- big data
- media discontinuity
- author-dependent

Challenges in Live Processing of the Writing Process

- non-linear
- revisions
- big data
- media discontinuity
- author-dependent

Challenges in Live Processing of the Writing Process

- non-linear
- revisions
- big data
- media discontinuity
- author-dependent

Challenges in Live Processing of the Writing Process

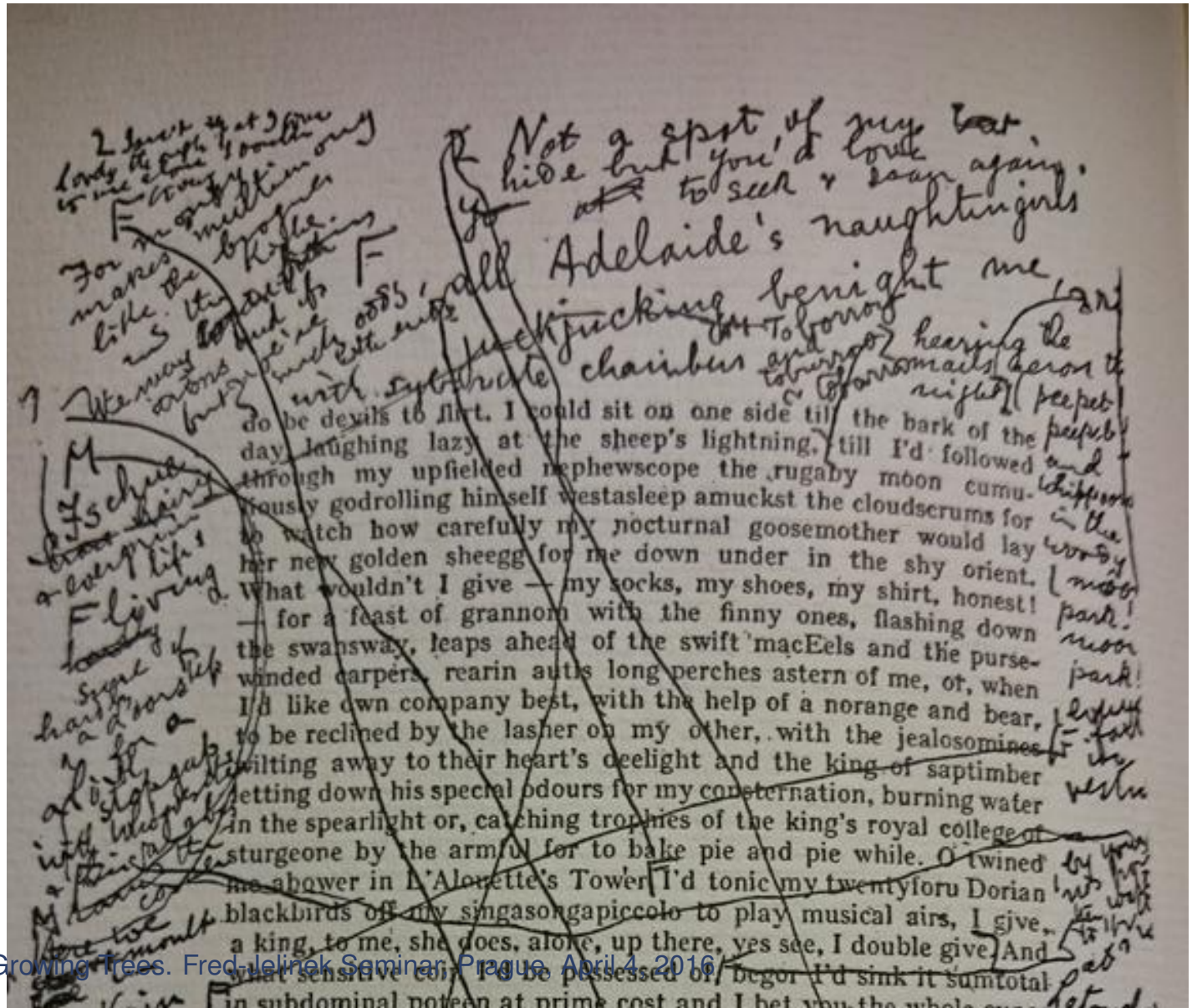
- non-linear
- revisions
- big data
- media discontinuity
- author-dependent

Challenges in Live Processing of the Writing Process

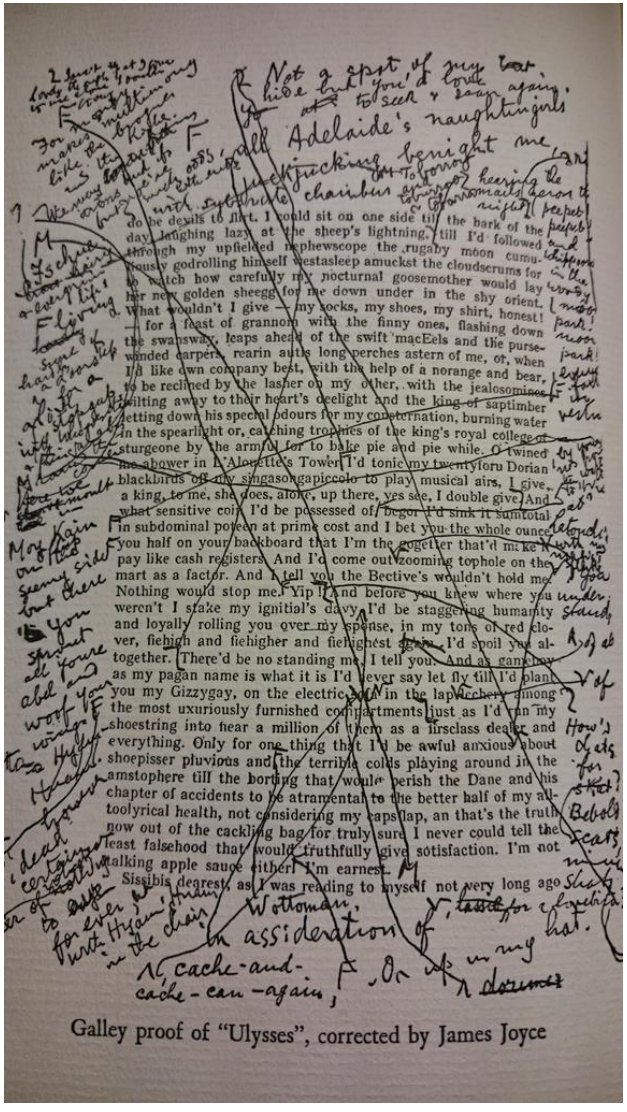
- non-linear
- revisions
- big data
- media discontinuity
- author-dependent

Media Discontinuity

Media Discontinuity



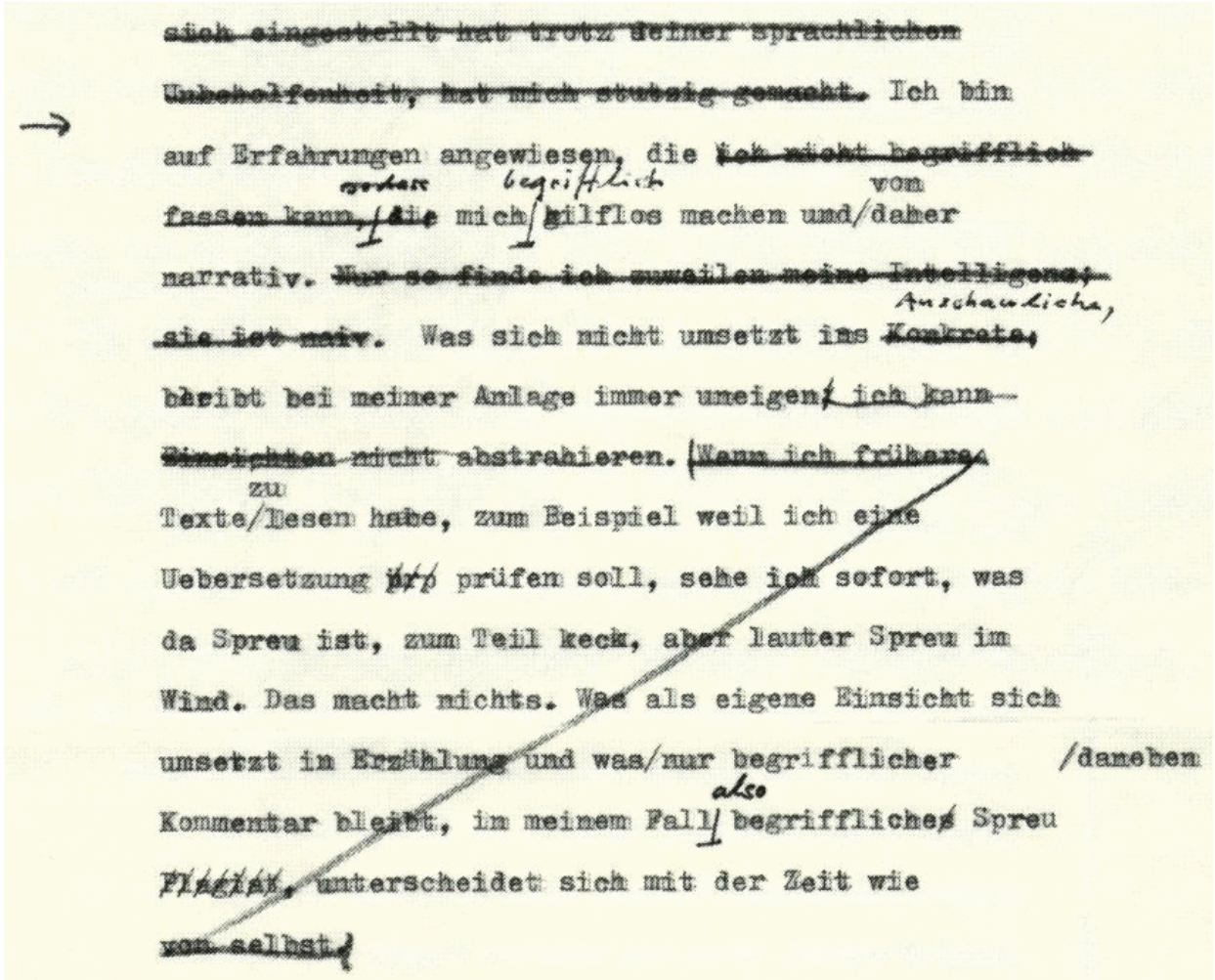
Media Discontinuity



Galley proof of "Ulysses", corrected by James Joyce

(James Joyce)

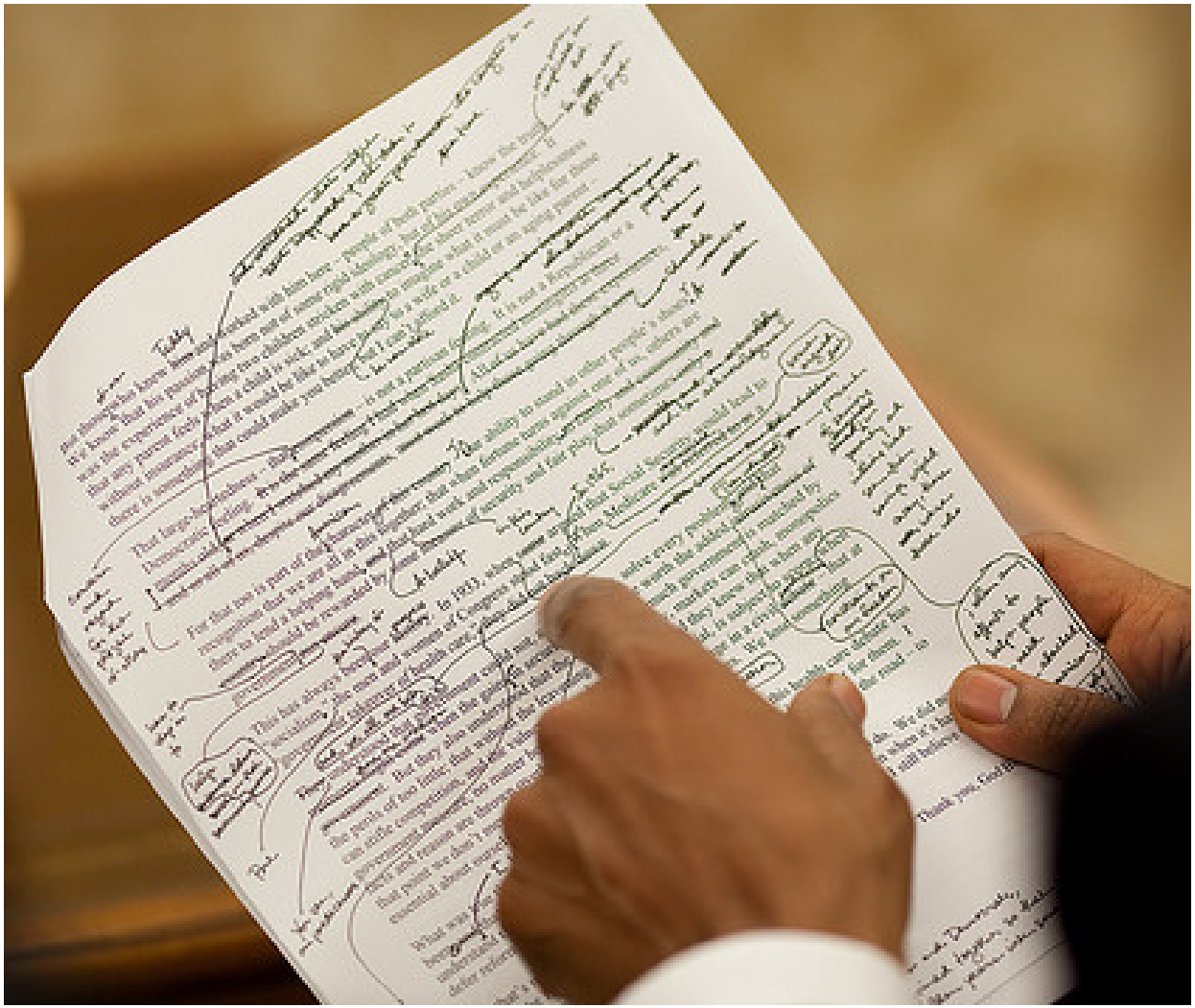
Media Discontinuity



sich eingeebnet hat trotz deiner sprachlichen
Unbeholfenheit, hat mich stutzig gemacht. Ich bin
auf Erfahrungen angewiesen, die ich nicht begrifflich
sondern begrifflich vom
fassen kann, die mich hilflos machen und daher
narrativ. Nur so finde ich zuweilen meine Intelligenz,
Anschauliche,
sie ich mein. Was sich nicht umsetzt ins Konkrete,
bleibt bei meiner Anlage immer uneigen, ich kann
Einsichten nicht abstrahieren. Wenn ich früheren
zu
Texte/Lesen habe, zum Beispiel weil ich eine
Uebersetzung prüfen soll, sehe ich sofort, was
da Spreu ist, zum Teil keck, aber lauter Spreu im
Wind. Das macht nichts. Was als eigene Einsicht sich
umsetzt im Erzählung und was/nur begrifflicher /daneben
also
Kommentar bleibt, im meinem Fall/ begriffliches Spreu
Flugsand, unterscheidet sich mit der Zeit wie
von selbst.

(Max Frisch)

Media Discontinuity



(Barack Obama)

Big Data

- Recording of the writing process by keystroke-logging
- Keystroke-logs grow as we type
- Size of keystroke-logs
  - Example: 2'130 data points for a 23 minutes writing session (1.3 MB XML data, 160 words final text)
  - Snapshot: Data for producing the German determiner "Die"

INSTITUT FÜR DEUTSCHE SPRACHE

Big Data

- Recording of the writing process by keystroke-logging
- Keystroke-logs grow as we type
- Size of keystroke-logs
  - Example: 2’130 data points for a 23 minutes writing session (1.3 MB XML data, 160 words final text)
  - Snapshot: Data for producing the German determiner “Die”

Big Data

- Recording of the writing process by keystroke-logging
- Keystroke-logs grow as we type
- Size of keystroke-logs
  - Example: 2'130 data points for a 23 minutes writing session (1.3 MB XML data, 160 words final text)
  - Snapshot: Data for producing the German determiner "Die"

Big Data

- Recording of the writing process by keystroke-logging
- Keystroke-logs grow as we type
- Size of keystroke-logs
  - Example: 2’130 data points for a 23 minutes writing session (1.3 MB XML data, 160 words final text)
  - Snapshot: Data for producing the German determiner “Die”

Big Data

- Recording of the writing process by keystroke-logging
- Keystroke-logs grow as we type
- Size of keystroke-logs
  - Example: 2’130 data points for a 23 minutes writing session (1.3 MB XML data, 160 words final text)
  - Snapshot: Data for producing the German determiner “Die”

Big Data

```
<event>
  <id>2</id>
  <type>keyboard</type>
  <output>D</output>
  <position>0</position>
  <positionFull>0</positionFull>
  <doclength>1</doclength>
  <doclengthFull>1</doclengthFull>
  <charProduction>1</charProduction>
  <startTime>1525434</startTime>
  <startClock>00:25:25.434</startClock>
  <endTime>1525494</endTime>
  <endClock>00:25:25.494</endClock>
  <actionTime>251</actionTime>
  <pauseTime>0</pauseTime>
  <pauseLocation>8</pauseLocation>
  <pauseLocationFull>INITIAL</pauseLocationFull>
</event>
<event>
  <id>3</id>
  <type>keyboard</type>
  <output>i</output>
  <position>1</position>
  <positionFull>1</positionFull>
  <doclength>2</doclength>
  <doclengthFull>2</doclengthFull>
  <charProduction>1</charProduction>
  <startTime>1525664</startTime>
  <startClock>00:25:25.664</startClock>
  <endTime>1525754</endTime>
  <endClock>00:25:25.754</endClock>
  <actionTime>90</actionTime>
  <pauseTime>230</pauseTime>
  <pauseLocation>1</pauseLocation>
  <pauseLocationFull>WITHIN WORDS</pauseLocationFull>
</event>
<event>
  <id>4</id>
  <type>keyboard</type>
  <output>e</output>
  <position>2</position>
  <positionFull>2</positionFull>
  <doclength>3</doclength>
  <doclengthFull>3</doclengthFull>
  <charProduction>2</charProduction>
  <startTime>1525774</startTime>
  <startClock>00:25:25.774</startClock>
  <endTime>1525824</endTime>
  <endClock>00:25:25.824</endClock>
  <actionTime>50</actionTime>
  <pauseTime>110</pauseTime>
  <pauseLocation>1</pauseLocation>
  <pauseLocationFull>WITHIN WORDS</pauseLocationFull>
</event>
```

(Producing “Die”)

Example Data

- Writing session from a seminar
- German
- Students
- Argumentative Essay:
  - 150 to 160 words
  - 25 minutes time
  - free topic (here: “Should assisted suicide be legalized?”)
- Original research hypothesis:
  - Restrictions on time and size trigger extensive revisions.
  - (proven only for size)
- Keystroke-Logging with Inputlog
  `www.inputlog.net`

Example Data

- Writing session from a seminar
- German
- Students
- Argumentative Essay:
  - 150 to 160 words
  - 25 minutes time
  - free topic (here: “Should assisted suicide be legalized?”)
- Original research hypothesis:
  - Restrictions on time and size trigger extensive revisions.
  - (proven only for size)
- Keystroke-Logging with Inputlog
  `www.inputlog.net`

Example Data

- Writing session from a seminar
- German
- Students
- Argumentative Essay:
  - 150 to 160 words
  - 25 minutes time
  - free topic (here: “Should assisted suicide be legalized?”)
- Original research hypothesis:
  - Restrictions on time and size trigger extensive revisions.
  - (proven only for size)
- Keystroke-Logging with Inputlog
  `www.inputlog.net`

Example Data

- Writing session from a seminar
- German
- Students
- Argumentative Essay:
  - 150 to 160 words
  - 25 minutes time
  - free topic (here: “Should assisted suicide be legalized?”)
- Original research hypothesis:
  - Restrictions on time and size trigger extensive revisions.
  - (proven only for size)
- Keystroke-Logging with Inputlog

  `www.inputlog.net`

Revisions and Non-Linearity

- Deletions and insertions at any time at any location.
- S-Notation for marking revisions.
- Snapshot: first part of the example text produced.

Die Legalisierun[h]1|1g [[akriver]2|2der aktiven]67|68 {aktiver}68|69 Sterbehilfe ist ein sehr zwiespältiges [thema]3|3Themas. [Für eine Legalisierung]69|70{Dafür}70|71 [w[ur]4|4ürde [S]5|5sprechen]71|72{spr[öche]73|74{äche}74|75}72|73, dass es totkran[kenmen]6|6ken Menschen ein sanftes Ab[blben]7|7leben ermöglich{t}40|41 und[[unötiges]8|8 w[ietere]9|9eiteres]76|77{ weiteres }77|78unnö[ä]10|10tiges Leid verhindert.

Revisions and Non-Linearity

- Deletions and insertions at any time at any location.
- S-Notation for marking revisions.
- Snapshot: first part of the example text produced.

Die Legalisierun[h]1|1g [[akriver]2|2der aktiven]67|68
{aktiver}68|69 Sterbehilfe ist ein sehr zwiespältiges
[thema]3|3Themas. [Für eine Legalisierung]69|70{Dafür}70|71
[w[ur]4|4ürde [S]5|5sprechen]71|72{spr[öche]73|74{äche}74|75}72
|73, dass es totkran[kenmen]6|6ken Menschen ein sanftes
Ab[blben]7|7leben ermöglich{t}40|41 und[[unötiges]8|8
w[ietere]9|9eiteres]76|77{ weiteres }77|78unnö[ä]10|10tiges
Leid verhindert.

Revisions and Non-Linearity

- Deletions and insertions at any time at any location.
- S-Notation for marking revisions.
- Snapshot: first part of the example text produced.

Die Legalisierun[h]1|1g [[akriver]2|2der aktiven]67|68
{aktiver}68|69 Sterbehilfe ist ein sehr zwiespältiges
[thema]3|3Themas. [Für eine Legalisierung]69|70{Dafür}70|71
[w[ur]4|4ürde [S]5|5sprechen]71|72{spr[öche]73|74{äche}74|75}72
|73, dass es totkran[kenmen]6|6ken Menschen ein sanftes
Ab[blben]7|7leben ermöglich{t}40|41 und[[unötiges]8|8
w[ietere]9|9eiteres]76|77{ weiteres }77|78unnö[ä]10|10tiges
Leid verhindert.

Revisions and Non-Linearity

- Deletions and insertions at any time at any location.
- S-Notation for marking revisions.
- Snapshot: first part of the example text produced.

Die Legalisierun[h]1|1g [[akriver]2|2der aktiven]67|68
{aktiver}68|69 Sterbehilfe ist ein sehr zwiespältiges
[thema]3|3Themas. [Für eine Legalisierung]69|70{Dafür}70|71
[w[ur]4|4ürde [S]5|5sprechen]71|72{spr[öche]73|74{äche}74|75}72
|73, dass es totkran[kenmen]6|6ken Menschen ein sanftes
Ab[blben]7|7leben ermöglich{t}40|41 und[[unötiges]8|8
w[ietere]9|9eiteres]76|77{ weiteres }77|78unnö[ä]10|10tiges
Leid verhindert.

Revisions and Non-Linearity

- Deletions and insertions at any time at any location.
- S-Notation for marking revisions.
- Snapshot: first part of the example text produced.

Die Legalisierun[h]1|1g [[akriver]2|2der aktiven]67|68
{aktiver}68|69 Sterbehilfe ist ein sehr zwiespältiges
[thema]3|3Themas. [Für eine Legalisierung]69|70{Dafür}70|71
[w[ur]4|4ürde [S]5|5sprechen]71|72{spr[öche]73|74{äche}74|75}72
|73, dass es totkran[kenmen]6|6ken Menschen ein sanftes
Ab[blben]7|7leben ermöglich{t}40|41 und[[unötiges]8|8
w[ietere]9|9eiteres]76|77{ weiteres }77|78unnö[ä]10|10tiges
Leid verhindert.

Revisions and Non-Linearity

- Deletions and insertions at any time at any location.
- S-Notation for marking revisions.
- Snapshot: first part of the example text produced.

```
Die Legalisierun[h]1|1g [[akriver]2|2der aktiven]67|68
{aktiver}68|69 Sterbehilfe ist ein sehr zwiespältiges
[thema]3|3Themas. [Für eine Legalisierung]69|70{Dafür}70|71
[w[ur]4|4ürde [S]5|5sprechen]71|72{spr[öche]73|74{äche}74|75}72
|73, dass es totkran[kenmen]6|6ken Menschen ein sanftes
Ab[blben ]7|7leben ermöglich{t}40|41 und[ [unötiges ]8|8
w[ietere]9|9eiteres ]76|77{ weiteres }77|78unnö[ä]10|10tiges
Leid verhindert.
```

Approaches for Processing Writing Data

- During writing, a text has various states until the final text:
  - One state per keystroke-log-event. (2130)
  - One state per saved draft. (3)
  - One state per revision. (97)
  - One state per change in production mode. (140)
- A “state” is a “version”.
- Apply NLP to growing text/versions.
  - Show information on current word.
  - Show information on current sentence.
  - Show diffs of versions.
- Visualize growing.

Approaches for Processing Writing Data

- During writing, a text has various states until the final text:
  - One state per keystroke-log-event. (2’130)
  - One state per saved draft. (3)
  - One state per revision. (97)
  - One state per change in production mode. (140)
- A “state” is a “version”.
- Apply NLP to growing text/versions.
  - Show information on current word.
  - Show information on current sentence.
  - Show diffs of versions.
- Visualize growing.

Approaches for Processing Writing Data

- During writing, a text has various states until the final text:
  - One state per keystroke-log-event. (2’130)
  - One state per saved draft. (3)
  - One state per revision. (97)
  - One state per change in production mode. (140)
- A “state” is a “version”.
- Apply NLP to growing text/versions.
  - Show information on current word.
  - Show information on current sentence.
  - Show diffs of versions.
- Visualize growing.

Approaches for Processing Writing Data

- During writing, a text has various states until the final text:
  - One state per keystroke-log-event. (2’130)
  - One state per saved draft. (3)
  - One state per revision. (97)
  - One state per change in production mode. (140)
- A “state” is a “version”.
- Apply NLP to growing text/versions.
  - Show information on current word.
  - Show information on current sentence.
  - Show diffs of versions.
- Visualize growing.

Approaches for Processing Writing Data

- During writing, a text has various states until the final text:
  - One state per keystroke-log-event. (2’130)
  - One state per saved draft. (3)
  - One state per revision. (97)
  - One state per change in production mode. (140)
- A “state” is a “version”.
- Apply NLP to growing text/versions.
  - Show information on current word.
  - Show information on current sentence.
  - Show diffs of versions.
- Visualize growing.

Approaches for Processing Writing Data

- During writing, a text has various states until the final text:
  - One state per keystroke-log-event. (2’130)
  - One state per saved draft. (3)
  - One state per revision. (97)
  - One state per change in production mode. (140)
- A “state” is a “version”.
- Apply NLP to growing text/versions.
  - Show information on current word.
  - Show information on current sentence.
  - Show diffs of versions.
- Visualize growing.

Approaches for Processing Writing Data

- During writing, a text has various states until the final text:
  - One state per keystroke-log-event. (2’130)
  - One state per saved draft. (3)
  - One state per revision. (97)
  - One state per change in production mode. (140)
- A “state” is a “version”.
- Apply NLP to growing text/versions.
  - Show information on current word.
  - Show information on current sentence.
  - Show diffs of versions.
- Visualize growing.

Approaches for Processing Writing Data

- During writing, a text has various states until the final text:
  - One state per keystroke-log-event. (2’130)
  - One state per saved draft. (3)
  - One state per revision. (97)
  - One state per change in production mode. (140)
- A “state” is a “version”.
- Apply NLP to growing text/versions.
  - Show information on current word.
  - Show information on current sentence.
  - Show diffs of versions.
- Visualize growing.

Approaches for Processing Writing Data

- During writing, a text has various states until the final text:
  - One state per keystroke-log-event. (2’130)
  - One state per saved draft. (3)
  - One state per revision. (97)
  - One state per change in production mode. (140)
- A “state” is a “version”.
- Apply NLP to growing text/versions.
  - Show information on current word.
  - Show information on current sentence.
  - Show diffs of versions.
- Visualize growing.

Approaches for Processing Writing Data

- During writing, a text has various states until the final text:
  - One state per keystroke-log-event. (2’130)
  - One state per saved draft. (3)
  - One state per revision. (97)
  - One state per change in production mode. (140)
- A “state” is a “version”.
- Apply NLP to growing text/versions.
  - Show information on current word.
  - Show information on current sentence.
  - Show diffs of versions.
- Visualize growing.

Approaches for Processing Writing Data

- During writing, a text has various states until the final text:
  - One state per keystroke-log-event. (2’130)
  - One state per saved draft. (3)
  - One state per revision. (97)
  - One state per change in production mode. (140)
- A “state” is a “version”.
- Apply NLP to growing text/versions.
  - Show information on current word.
  - Show information on current sentence.
  - Show diffs of versions.
- Visualize growing.

Approaches for Processing Writing Data

- During writing, a text has various states until the final text:
  - One state per keystroke-log-event. (2’130)
  - One state per saved draft. (3)
  - One state per revision. (97)
  - One state per change in production mode. (140)
- A “state” is a “version”.
- Apply NLP to growing text/versions.
  - Show information on current word.
  - Show information on current sentence.
  - Show diffs of versions.
- Visualize growing.

Change in Production Mode

- Switch from:
  - normal text production to deletion or insertion
  - deletion to insertion or normal text production
  - insertion to deletion or normal text production
- “normal text production” is typing at the edge of the text produced so far.
- ↪ Operationalization of “version”:
  - Previous version can be accessed by executing `undo`.

Change in Production Mode

- Switch from:
  - normal text production to deletion or insertion
  - deletion to insertion or normal text production
  - insertion to deletion or normal text production
- “normal text production” is typing at the edge of the text produced so far.
- ↪ Operationalization of “version”:
  - Previous version can be accessed by executing `undo`.

Change in Production Mode

- Switch from:
  - normal text production to deletion or insertion
  - deletion to insertion or normal text production
  - insertion to deletion or normal text production
- “normal text production” is typing at the edge of the text produced so far.
- ↪ Operationalization of “version”:
  - Previous version can be accessed by executing `undo`.

Change in Production Mode

- Switch from:
  - normal text production to deletion or insertion
  - deletion to insertion or normal text production
  - insertion to deletion or normal text production
- “normal text production” is typing at the edge of the text produced so far.
- ↪ Operationalization of “version”:
  - Previous version can be accessed by executing `undo`.

Change in Production Mode

- Switch from:
  - normal text production to deletion or insertion
  - deletion to insertion or normal text production
  - insertion to deletion or normal text production
- “normal text production” is typing at the edge of the text produced so far.
- ↪ Operationalization of “version”:
  - Previous version can be accessed by executing `undo`.

Change in Production Mode

- Switch from:
  - normal text production to deletion or insertion
  - deletion to insertion or normal text production
  - insertion to deletion or normal text production
- “normal text production” is typing at the edge of the text produced so far.

↪ Operationalization of “version”:

- Previous version can be accessed by executing `undo`.

Change in Production Mode

- Switch from:
  - normal text production to deletion or insertion
  - deletion to insertion or normal text production
  - insertion to deletion or normal text production
- “normal text production” is typing at the edge of the text produced so far.

➥ Operationalization of “version”:

- Previous version can be accessed by executing `undo`.

Example Text (first 7 versions)

Example Text (first 7 versions)

(explicit versions)

```
Die Legalisierunh
Die Legalisierun
Die Legalisierung akriver
Die Legalisierung
Die Legalisierung der aktiven Sterbehilfe ist ein sehr
zwiespältiges thema
Die Legalisierung der aktiven Sterbehilfe ist ein sehr
zwiespältiges
Die Legalisierung der aktiven Sterbehilfe ist ein sehr
zwiespältiges Themas.  Für eine Legalisierung wur
```

Example Text (first 7 versions)

(S-notation)

```
Die Legalisierun[h]1|1g [akriver]2|2der aktiven
Sterbehilfe ist ein sehr zwiespältiges [thema]3|3Themas.
Für eine Legalisierung wur|4
```

Visualize growing text

- Use versioning/diffing algorithms from document engineering.
  - rely on explicit structure.
  - focus on documents, not on text.
- Use NLP parsing.
  - implicit structure of natural language sentences.
  - sentences during writing are ill-formed, incomplete, inconsistent.
  - when to parse?
  - how to diff parse-trees?
  - we need interactive parsing!

Visualize growing text

- Use versioning/diffing algorithms from document engineering.
  - rely on explicit structure.
  - focus on documents, not on text.
- Use NLP parsing.
  - implicit structure of natural language sentences.
  - sentences during writing are ill-formed, incomplete, inconsistent.
  - when to parse?
  - how to diff parse-trees?
  - we need interactive parsing!

Visualize growing text

- Use versioning/diffing algorithms from document engineering.
  - rely on explicit structure.
  - focus on documents, not on text.
- Use NLP parsing.
  - implicit structure of natural language sentences.
  - sentences during writing are ill-formed, incomplete, inconsistent.
  - when to parse?
  - how to diff parse-trees?
  - we need interactive parsing!

Visualize growing text

- Use versioning/diffing algorithms from document engineering.
  - rely on explicit structure.
  - focus on documents, not on text.
- Use NLP parsing.
  - implicit structure of natural language sentences.
  - sentences during writing are ill-formed, incomplete, inconsistent.
  - when to parse?
  - how to diff parse-trees?
  - we need interactive parsing!

Visualize growing text

- Use versioning/diffing algorithms from document engineering.
  - rely on explicit structure.
  - focus on documents, not on text.
- Use NLP parsing.
  - implicit structure of natural language sentences.
  - sentences during writing are ill-formed, incomplete, inconsistent.
  - when to parse?
  - how to diff parse-trees?
  - we need interactive parsing!

Visualize growing text

- Use versioning/diffing algorithms from document engineering.
  - rely on explicit structure.
  - focus on documents, not on text.
- Use NLP parsing.
  - implicit structure of natural language sentences.
  - sentences during writing are ill-formed, incomplete, inconsistent.
  - when to parse?
  - how to diff parse-trees?
  - we need interactive parsing!

Automatic Re-Parsing

- Avoid re-parsing untouched sentences.
- Avoid constant re-parsing
  (e.g., after each key pressed or every 10 seconds).
- Re-use parsed clauses/sentences that are moved around.
- Possibilities:
  - parse touched sentences when a version is saved (manually or automatically).
  - parse touched sentence after a change in production mode.
  - parse current sentence after a change in production mode.
  - parse everything after a change in production mode.

Automatic Re-Parsing

- Avoid re-parsing untouched sentences.
- Avoid constant re-parsing
  (e.g., after each key pressed or every 10 seconds).
- Re-use parsed clauses/sentences that are moved around.
- Possibilities:
  - parse touched sentences when a version is saved (manually or automatically).
  - parse touched sentence after a change in production mode.
  - parse current sentence after a change in production mode.
  - parse everything after a change in production mode.

Automatic Re-Parsing

- Avoid re-parsing untouched sentences.
- Avoid constant re-parsing
  (e.g., after each key pressed or every 10 seconds).
- Re-use parsed clauses/sentences that are moved around.
- Possibilities:
  - parse touched sentences when a version is saved (manually or automatically).
  - parse touched sentence after a change in production mode.
  - parse current sentence after a change in production mode.
  - parse everything after a change in production mode.

Automatic Re-Parsing

- Avoid re-parsing untouched sentences.
- Avoid constant re-parsing
  (e.g., after each key pressed or every 10 seconds).
- Re-use parsed clauses/sentences that are moved around.
- Possibilities:
  - parse touched sentences when a version is saved (manually or automatically).
  - parse touched sentence after a change in production mode.
  - parse current sentence after a change in production mode.
  - parse everything after a change in production mode.

Automatic Re-Parsing

- Avoid re-parsing untouched sentences.
- Avoid constant re-parsing
  (e.g., after each key pressed or every 10 seconds).
- Re-use parsed clauses/sentences that are moved around.
- Possibilities:
  - parse touched sentences when a version is saved (manually or automatically).
  - parse touched sentence after a change in production mode.
  - parse current sentence after a change in production mode.
  - parse everything after a change in production mode.

Automatic Re-Parsing

- Avoid re-parsing untouched sentences.
- Avoid constant re-parsing
  (e.g., after each key pressed or every 10 seconds).
- Re-use parsed clauses/sentences that are moved around.
- Possibilities:
  - parse touched sentences when a version is saved (manually or automatically).
  - parse touched sentence after a change in production mode.
  - parse current sentence after a change in production mode.
  - parse everything after a change in production mode.

Automatic Re-Parsing

- Avoid re-parsing untouched sentences.
- Avoid constant re-parsing
  (e.g., after each key pressed or every 10 seconds).
- Re-use parsed clauses/sentences that are moved around.
- Possibilities:
  - parse touched sentences when a version is saved (manually or automatically).
  - parse touched sentence after a change in production mode.
  - parse current sentence after a change in production mode.
  - parse everything after a change in production mode.

Automatic Re-Parsing

- Avoid re-parsing untouched sentences.
- Avoid constant re-parsing
  (e.g., after each key pressed or every 10 seconds).
- Re-use parsed clauses/sentences that are moved around.
- Possibilities:
  - parse touched sentences when a version is saved (manually or automatically).
  - parse touched sentence after a change in production mode.
  - parse current sentence after a change in production mode.
  - parse everything after a change in production mode.

Automatic Re-Parsing

- Avoid re-parsing untouched sentences.
- Avoid constant re-parsing
  (e.g., after each key pressed or every 10 seconds).
- Re-use parsed clauses/sentences that are moved around.
- Possibilities:
  - parse touched sentences when a version is saved (manually or automatically).
  - parse touched sentence after a change in production mode.
  - parse current sentence after a change in production mode.
  - parse everything after a change in production mode.

Interactive NLP in Text Editors

- Emacs as test bed.
- Information on current word.
  (Stripey Zebra implementation in Malaga for German morphology)
- Information on POS structures (aka “syntax highlighting”).
  (MBT for German)
- Information on syntax structure of current version.
  (Mate for German)

Interactive NLP in Text Editors

- Emacs as test bed.
- Information on current word.
  (Stripey Zebra implementation in Malaga for German morphology)
- Information on POS structures (aka “syntax highlighting”).
  (MBT for German)
- Information on syntax structure of current version.
  (Mate for German)

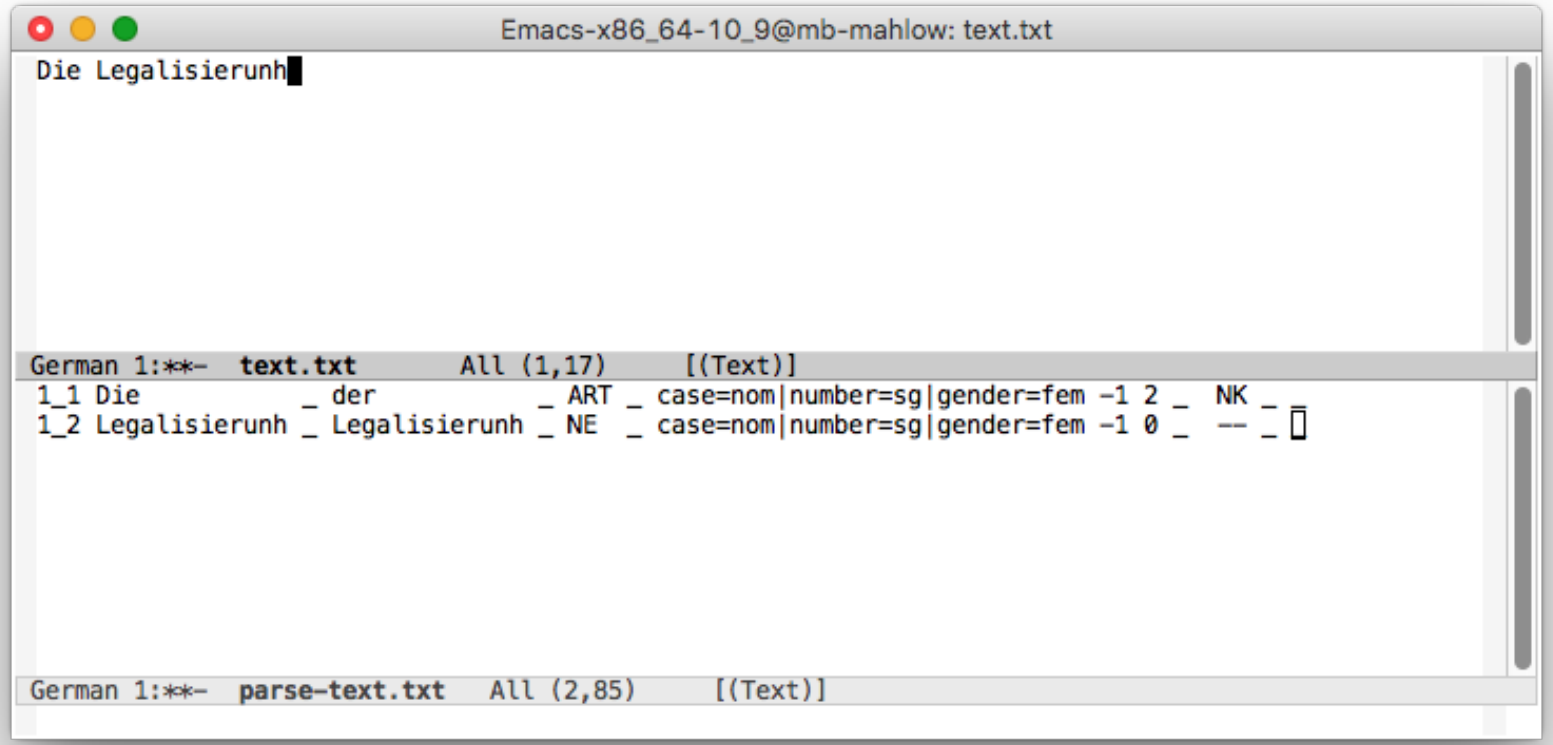
Interactive NLP in Text Editors

- Emacs as test bed.
- Information on current word.
  (Stripey Zebra implementation in Malaga for German morphology)
- Information on POS structures (aka “syntax highlighting”).
  (MBT for German)
- Information on syntax structure of current version.
  (Mate for German)

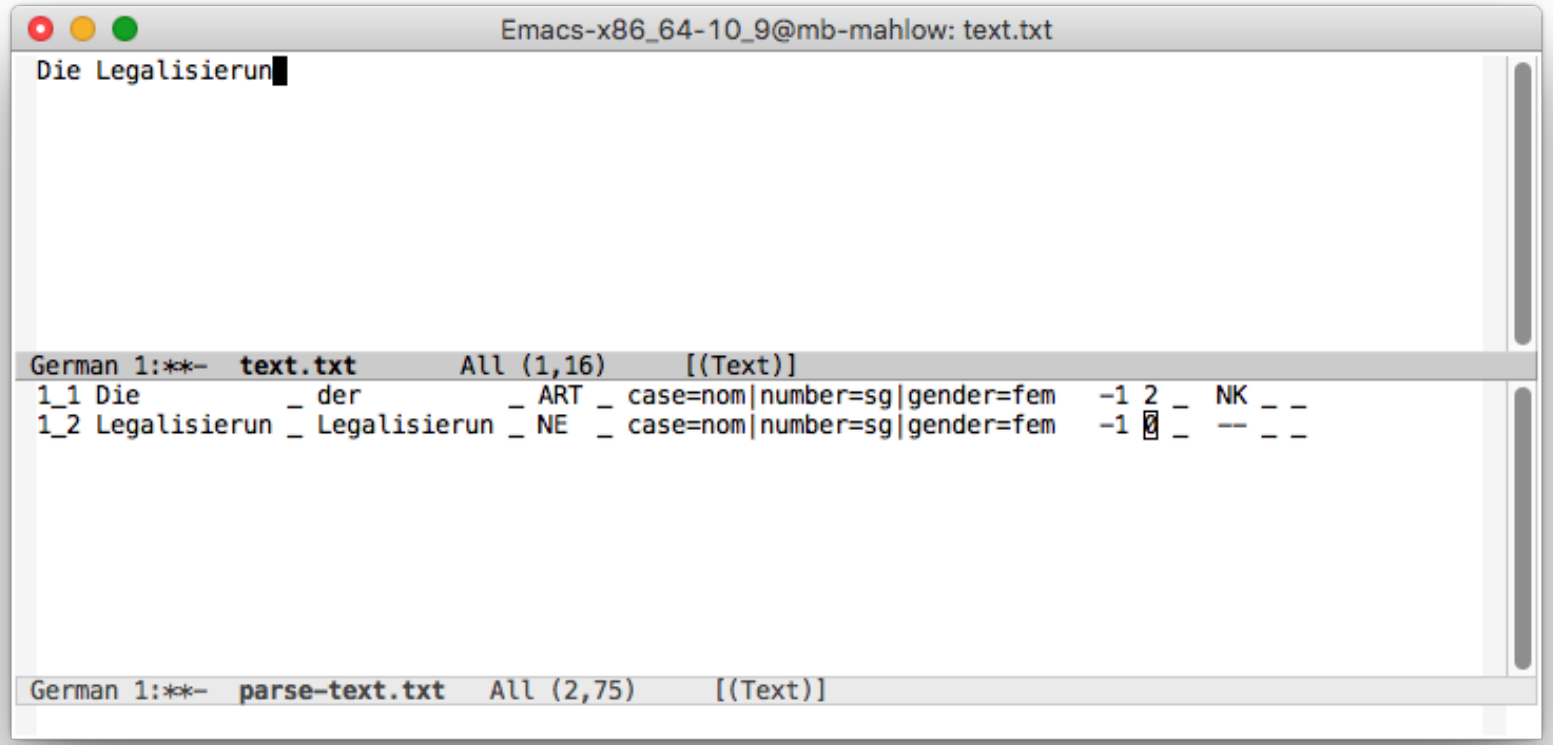
Interactive NLP in Text Editors

- Emacs as test bed.
- Information on current word.
  (Stripey Zebra implementation in Malaga for German morphology)
- Information on POS structures (aka “syntax highlighting”).
  (MBT for German)
- Information on syntax structure of current version.
  (Mate for German)

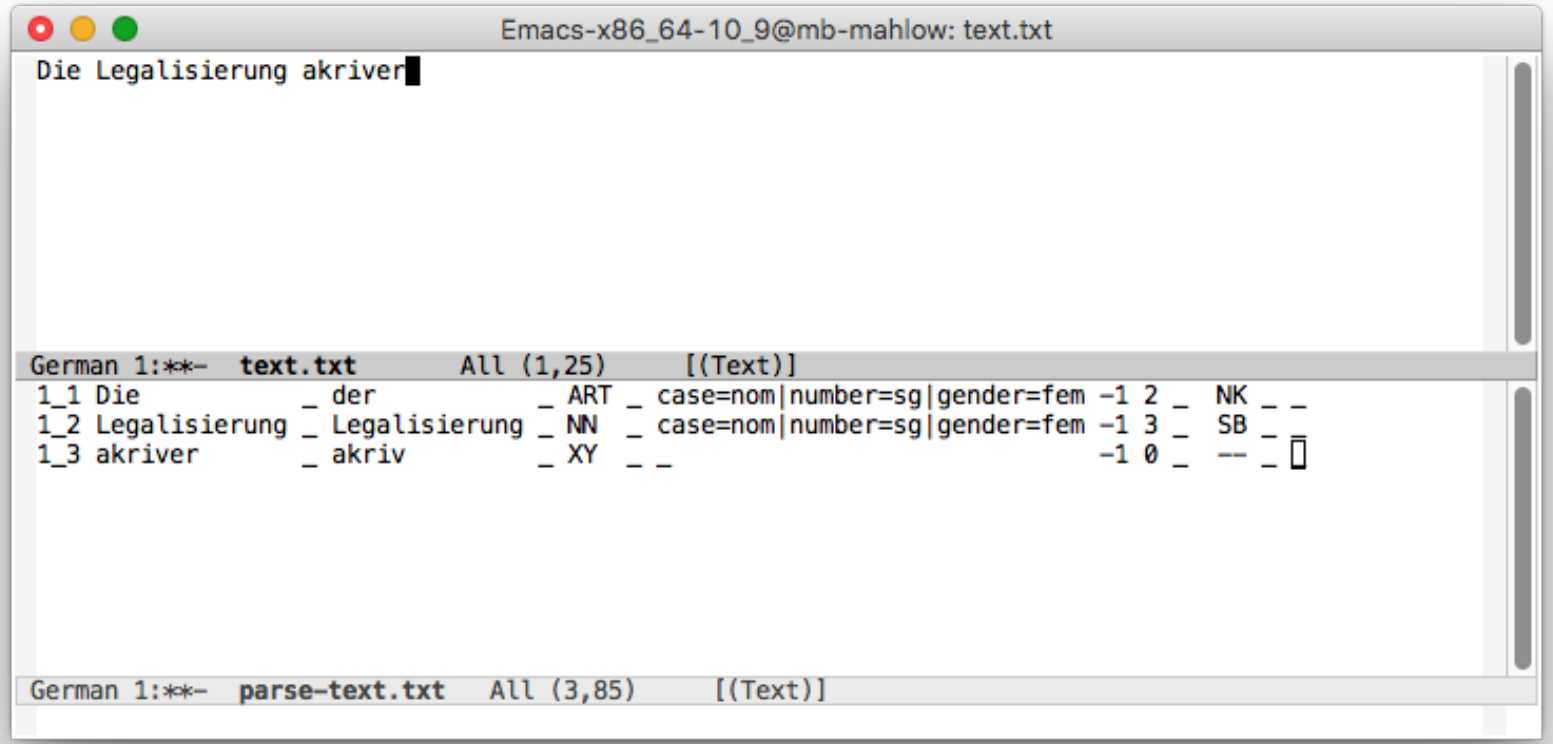
Parsing Versions



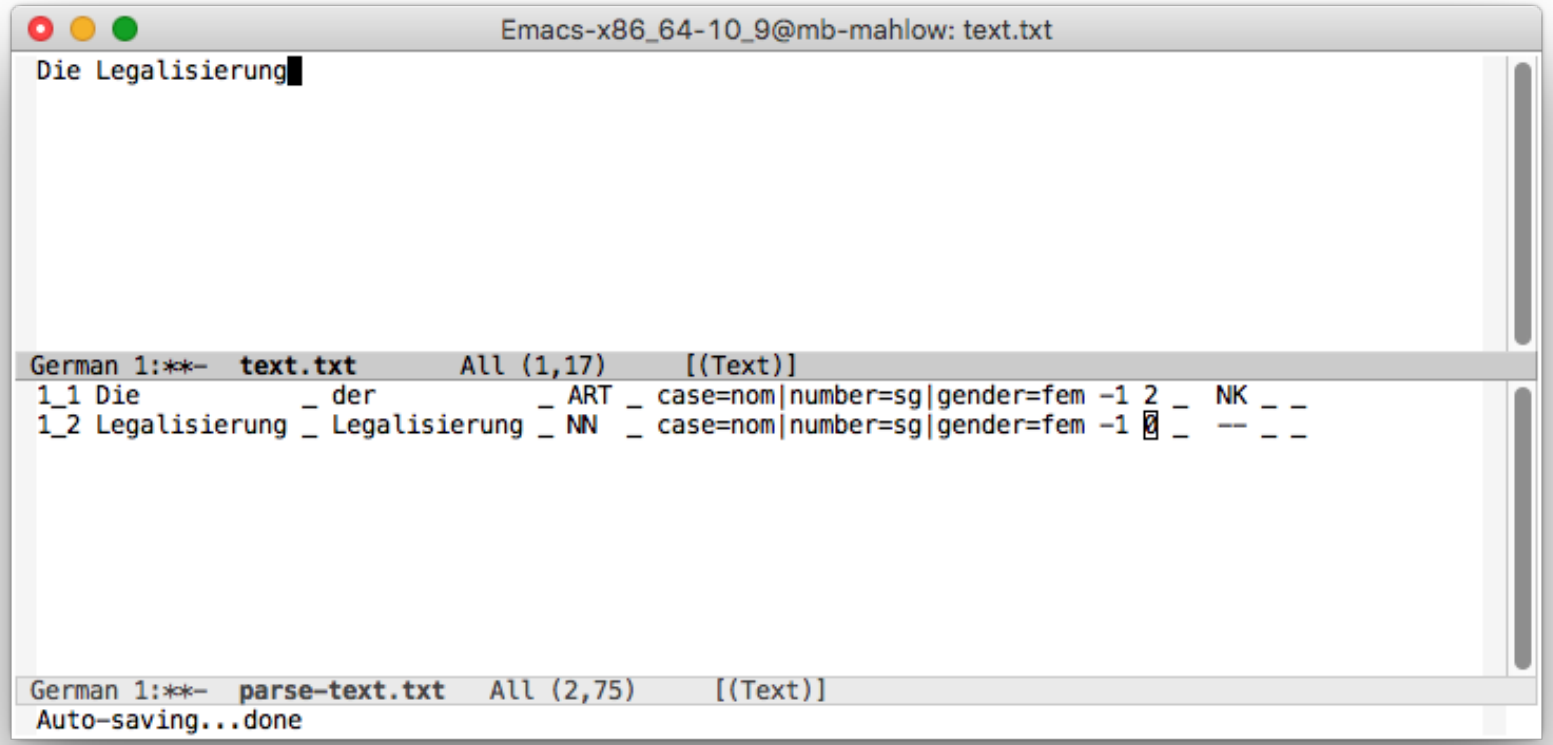
Parsing Versions



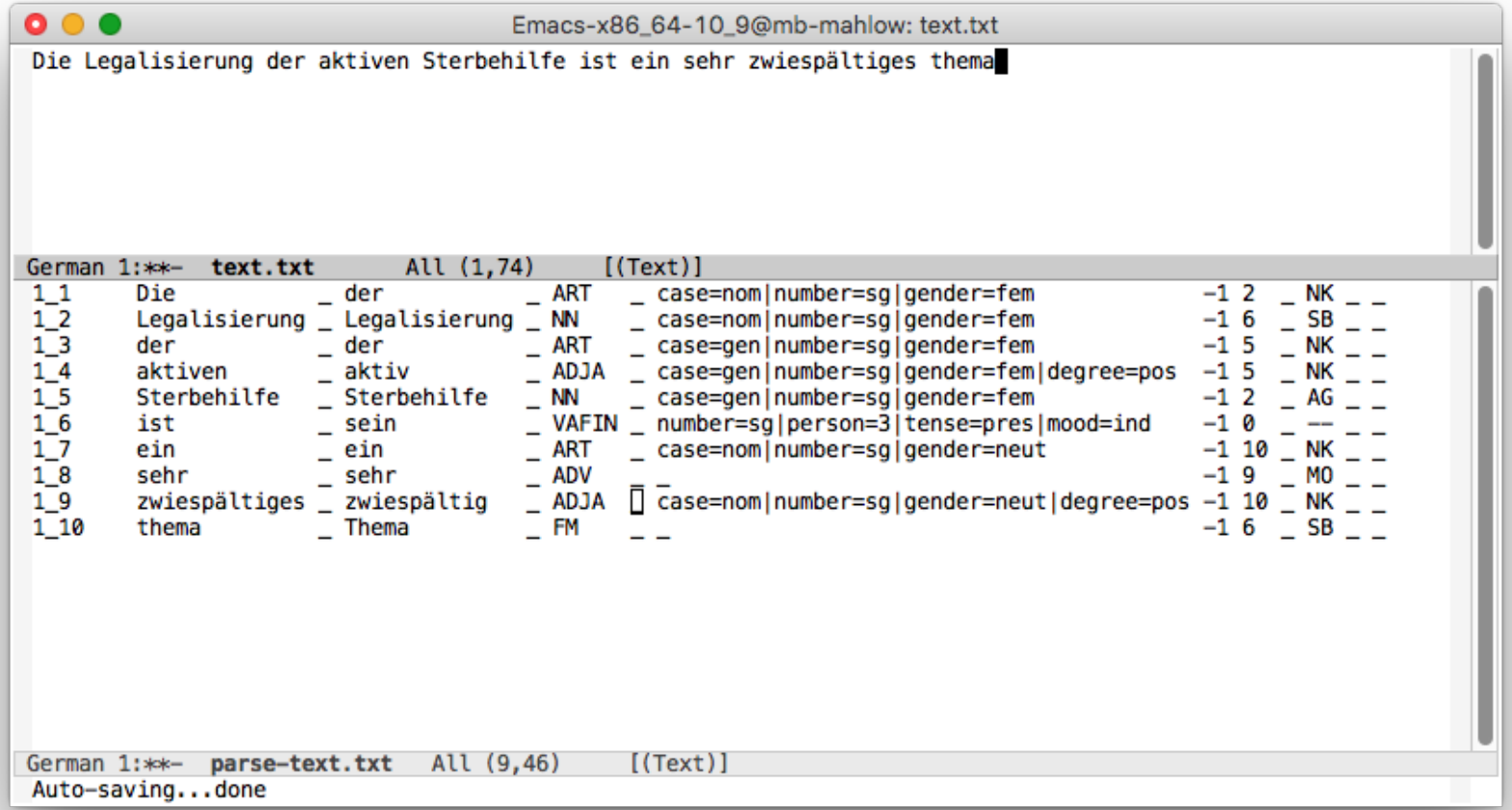
Parsing Versions



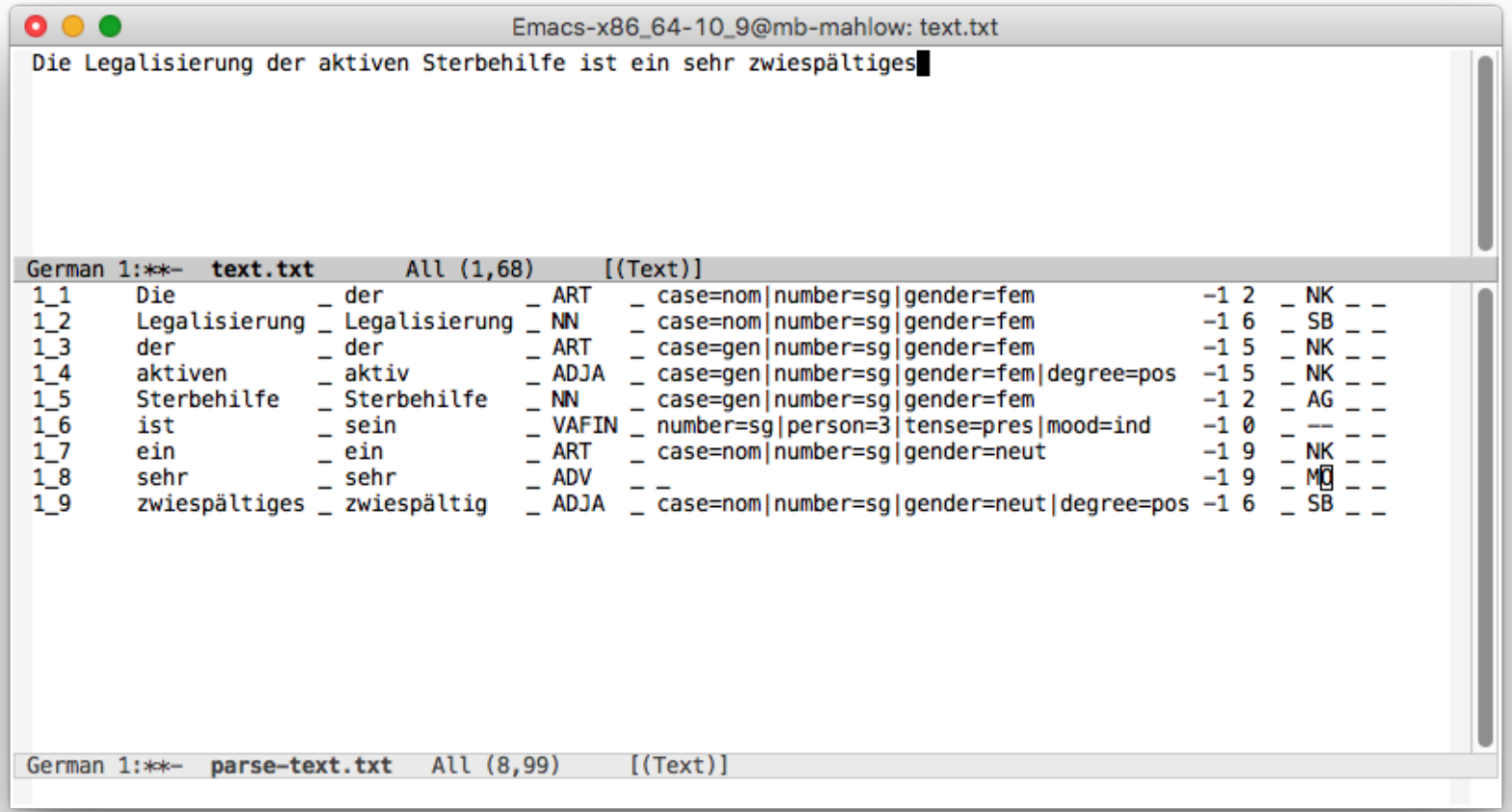
Parsing Versions



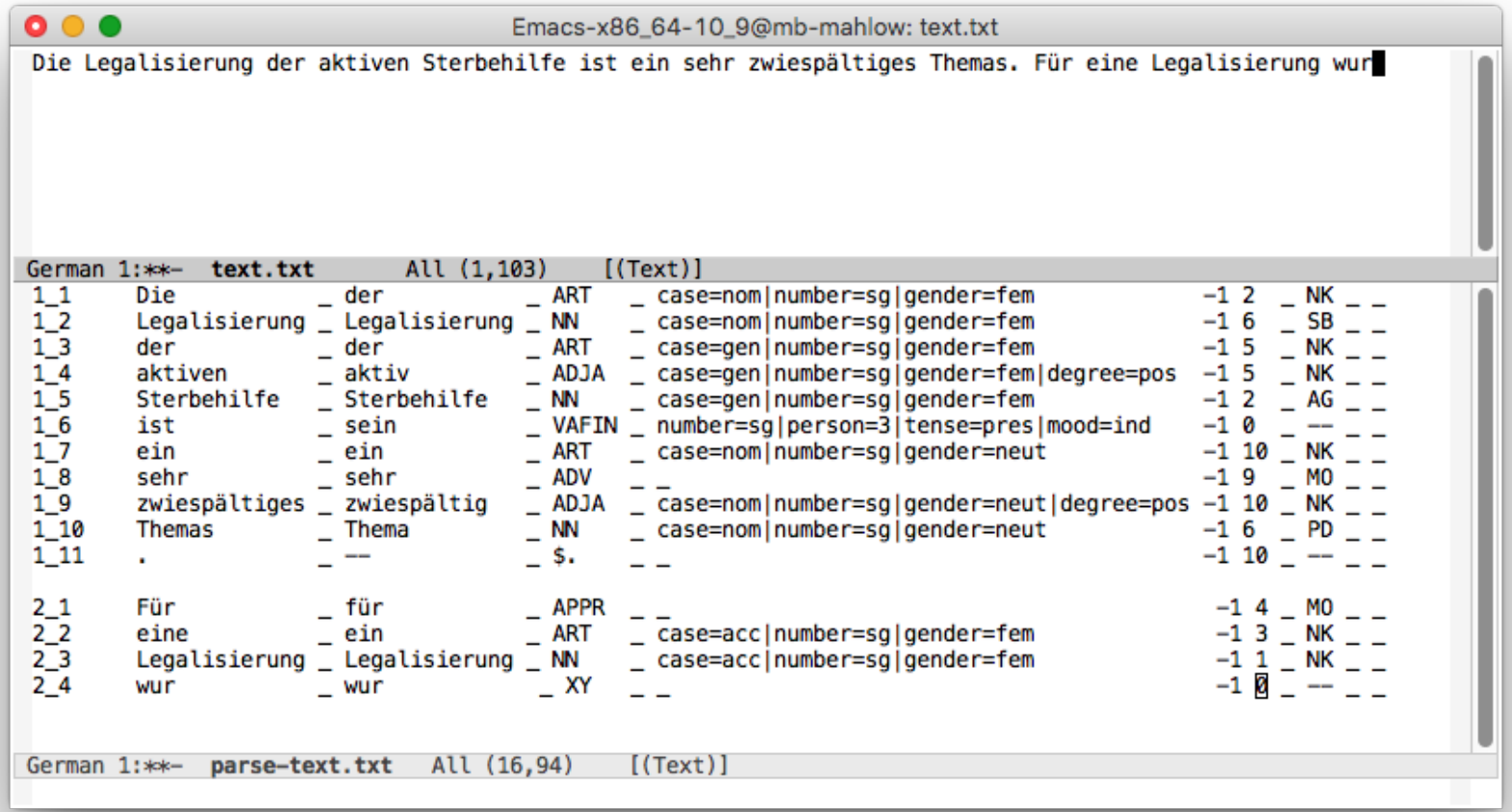
Parsing Versions



Parsing Versions



Parsing Versions



Summary

- Non-linear incremental parsing as parsing of text during production.
- ➥ Application of CS-style incremental parsing to natural language text “coding”.
- Static interactive NLP available on word and sentence level.

Summary

- Non-linear incremental parsing as parsing of text during production.
- ➥ Application of CS-style incremental parsing to natural language text “coding”.
- Static interactive NLP available on word and sentence level.

Summary

- Non-linear incremental parsing as parsing of text during production.
- ➥ Application of CS-style incremental parsing to natural language text “coding”.
- Static interactive NLP available on word and sentence level.

Open Challenges

- How to display parse trees to be useful for a writer?
- How to diff parse trees and display the diff?
- Apply structured editing.
- Use our version definition for scholarly editions, too.

Open Challenges

- How to display parse trees to be useful for a writer?
- How to diff parse trees and display the diff?
- Apply structured editing.
- Use our version definition for scholarly editions, too.

Open Challenges

- How to display parse trees to be useful for a writer?
- How to diff parse trees and display the diff?
- Apply structured editing.
- Use our version definition for scholarly editions, too.

Growing Trees:
Non-Linear Incremental Parsing during Writing

Cerstin Mahlow
Institut für Deutsche Sprache
mahlow@ids-mannheim.de

Fred-Jelinek Seminar, Prague, April 4, 2016