

# 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

- ▶ on-going process during reading
- ▶ 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



# 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



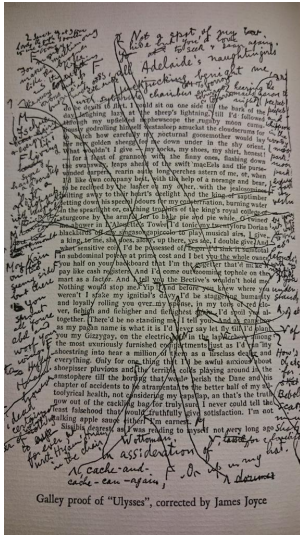


INSTITUT FÜR  
DEUTSCHE SPRACHE

5/19



# Media Discontinuity



Galley proof of "Ulysses", corrected by James Joyce

(James Joyce)



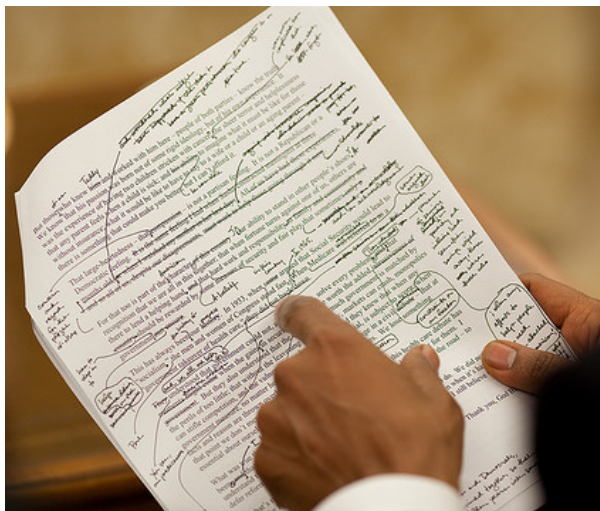
# Media Discontinuity

→  
~~sich eingestellt hat trotz seiner sprachlichen~~  
~~Unbehelfenheit, hat mich stutzig gemacht.~~ Ich bin  
auf Erfahrungen angewiesen, die ~~ich nicht begrifflich~~  
<sup>vorher</sup> ~~fassen kann,~~ <sup>begrifflich</sup> ~~die mich hilflos machen und daher~~ <sup>von</sup>  
narrativ. ~~Nur so finde ich zuweilen meine Intelligenz,~~  
~~sie ist nativ.~~ <sup>Anschauliche,</sup> Was sich nicht umsetzt ins ~~Konkrete,~~  
bleibt bei meiner Anlage immer uneigen, ~~ich kann-~~  
~~Einsichten nicht abstrahieren.~~ ~~(Wenn ich friheres~~  
<sup>zu</sup> Texte/lesen habe, zum Beispiel weil ich eine  
Übersetzung ~~prüfen~~ 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 in Erzählung und was/nur begrifflicher <sup>/daneben</sup>  
Kommentar bleibt, in meinem Fall <sup>also</sup> ~~begriffliches~~ Spreu  
~~Platz~~, 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: 2130 data points for a 23 minutes writing session (1.3 MB XML data, 100 words final text)
  - ▶ Graphical Data by producing the German Dictionary "Duden"



# 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>0</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>0</pauseLocation>
  <pauseLocationFull>INITIAL</pauseLocationFull>
</event>
<event>
  <id>3</id>
  <type>keyboard</type>
  <output>1</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](http://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](http://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](http://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](http://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 Legalisierung[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}75  
|73, dass es totkran[kenmen]6|6ken Menschen ein sanftes  
Ab[blben ]7|7leben ermöglichen{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 Legalisierung[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 ermöglichen{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 Legalisierung[h]1|1g [[aktiver]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 ermöglichen{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 Legalisierung[h]1|1g [[aktiver]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 ermöglichen{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 Legalisierung[h]1|1g [[aktiver]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 ermöglichen{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 Legalisierung[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 ermöglichen{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. (24300)
  - ▶ One state per saved draft. (3)
  - ▶ One state per revision. (37)
  - ▶ 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 data on paragraphs.
- ▶ 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 Legalisierung[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 untouched sentences when a version is added (manually or automatically).
  - parse untouched sentences 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)

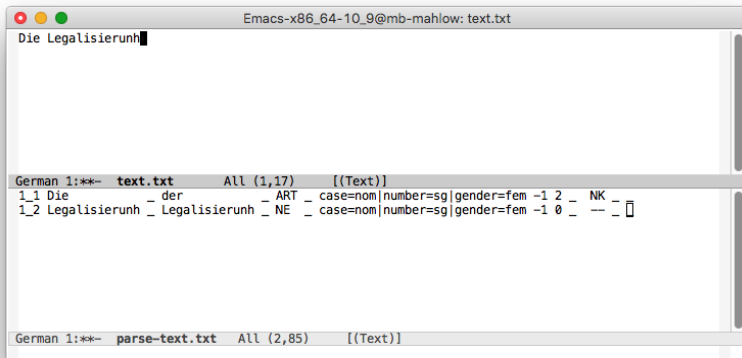


# Interactive NLP in Text Editors

- ▶ 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



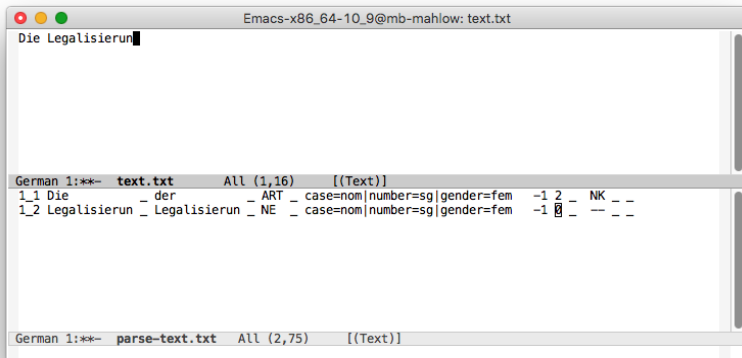
The screenshot shows an Emacs window titled "Emacs-x86\_64-10\_9@mb-mahlow: text.txt". The main text area contains the sentence "Die Legalisierung". Below the text, a parse tree is displayed for the sentence. The parse tree is shown in a buffer named "German 1:\*\*\*- text.txt" at position (1,17). The tree structure is as follows:

```
German 1:***- text.txt All (1,17) [(Text)]
1_1 Die _ der _ ART _ case=nom|number=sg|gender=fem -1 2 _ NK _
1_2 Legalisierung _ Legalisierung _ NE _ case=nom|number=sg|gender=fem -1 0 _ -- _
```

Below the parse tree, another buffer is shown: "German 1:\*\*\*- parse-text.txt" at position (2,85), which contains the text "[(Text)]".



# Parsing Versions



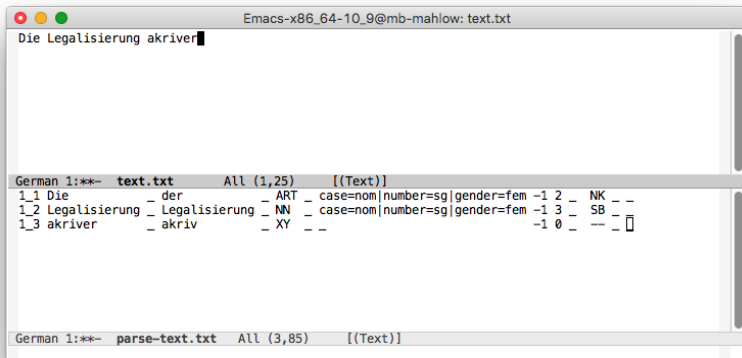
The screenshot shows an Emacs window titled "Emacs-x86\_64-10\_9@mb-mahlow: text.txt". The main text area contains the sentence "Die Legalisierung". Below the text, a parse tree is displayed. The tree has a root node "German 1:\*\*- text.txt" with children "All (1,16)" and "[[Text]]". The "All (1,16)" node has two children: "1\_1 Die" and "1\_2 Legalisierung". The "1\_1 Die" node has children "der" and "ART", with attributes "case=nom|number=sg|gender=fem". The "1\_2 Legalisierung" node has children "Legalisierung" and "NE", with attributes "case=nom|number=sg|gender=fem". The "NE" node has a child "1\_2" with attributes "NK" and "--".

```
German 1:**- text.txt All (1,16) [[Text]]
1_1 Die _ der _ ART _ case=nom|number=sg|gender=fem -1 2 _ NK _ _
1_2 Legalisierung _ Legalisierung _ NE _ case=nom|number=sg|gender=fem -1 2 _ -- _ _
```

German 1:\*\*- parse-text.txt All (2,75) [[Text]]



# Parsing Versions



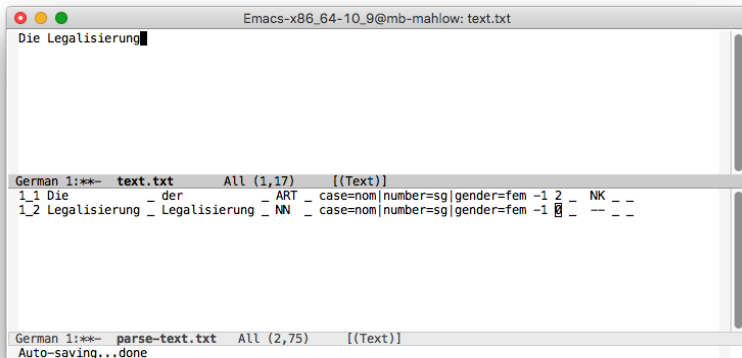
The screenshot shows an Emacs window titled "Emacs-x86\_64-10\_9@mb-mahlow: text.txt". The main text area contains the sentence "Die Legalisierung akriver". Below the text, a parse tree is displayed, showing the hierarchical structure of the sentence. The parse tree is organized into sections, with the first section labeled "German 1:\*\*- text.txt All (1,25) [(Text)]". The tree structure is as follows:

```
German 1:**- text.txt All (1,25) [(Text)]
├── 1_1 Die
│   └── der
│       └── ART
│           └── case=nom|number=sg|gender=fem -1 2 _ NK _ _
├── 1_2 Legalisierung
│   └── Legalisierung
│       └── NN
│           └── case=nom|number=sg|gender=fem -1 3 _ SB _ _
└── 1_3 akriver
    └── akviv
        └── XY
            └── -1 0 _ -- _ _
```

The second section of the parse tree is labeled "German 1:\*\*- parse-text.txt All (3,85) [(Text)]".



# Parsing Versions



The screenshot shows an Emacs window titled "Emacs-x86\_64-10\_9@mb-mahlow: text.txt". The main text area contains the sentence "Die Legalisierung". Below the text, there are two parse tree views. The first view is for the text "text.txt" and shows a tree structure for the sentence. The second view is for the text "parse-text.txt" and shows a tree structure for the same sentence. The status bar at the bottom indicates "Auto-saving...done".

```
German 1:**- text.txt All (1,17) [(Text)]
1_1 Die _ der _ ART _ case=nom|number=sg|gender=fem -1 2 _ NK _ _
1_2 Legalisierung _ Legalisierung _ NN _ case=nom|number=sg|gender=fem -1 0 _ -- _ _

German 1:**- parse-text.txt All (2,75) [(Text)]
Auto-saving...done
```



# Parsing Versions

```
Emacs-x86_64-10.9@mb-mahlow: text.txt
Die Legalisierung der aktiven Sterbehilfe ist ein sehr zwiespältiges thema
```

---

```
German 1:**- text.txt All (1,74) [(Text)]
1_1 Die _ der _ ART _ case=nom|number=sg|gender=fem -1 2 _ NK _
1_2 Legalisierung _ Legalisierung _ NN _ case=nom|number=sg|gender=fem -1 6 _ SB _
1_3 der _ der _ ART _ case=gen|number=sg|gender=fem -1 5 _ NK _
1_4 aktiven _ aktiv _ ADJA _ case=gen|number=sg|gender=fem|degree=pos -1 5 _ NK _
1_5 Sterbehilfe _ Sterbehilfe _ NN _ case=gen|number=sg|gender=fem -1 2 _ AG _
1_6 ist _ sein _ VAFIN _ number=sg|person=3|tense=pres|mood=ind -1 0 _ _ _
1_7 ein _ ein _ ART _ case=nom|number=sg|gender=neut -1 10 _ NK _
1_8 sehr _ sehr _ ADV _ -1 9 _ MO _
1_9 zwiespältiges _ zwiespältig _ ADJA _ case=nom|number=sg|gender=neut|degree=pos -1 10 _ NK _
1_10 thema _ Thema _ FM _ -1 6 _ SB _
```

---

```
German 1:**- parse-text.txt All (9,46) [(Text)]
Auto-saving...done
```



# Parsing Versions

Emacs-x86\_64-10.9@mb-mahlow: text.txt

Die Legalisierung der aktiven Sterbehilfe ist ein sehr zwiespältiges

German 1:**	text.txt	All (1,68)	[(Text)]
1_1	Die	der	ART case=nom number=sg gender=fem -1 2 NK
1_2	Legalisierung	Legalisierung	NN case=nom number=sg gender=fem -1 6 SB
1_3	der	der	ART case=gen number=sg gender=fem -1 5 NK
1_4	aktiven	aktiv	ADJA case=gen number=sg gender=fem degree=pos -1 5 NK
1_5	Sterbehilfe	Sterbehilfe	NN case=gen number=sg gender=fem -1 2 AG
1_6	ist	sein	VAFIN number=sg person=3 tense=pres mood=ind -1 0
1_7	ein	ein	ART case=nom number=sg gender=neut -1 9 NK
1_8	sehr	sehr	ADV -1 9 M
1_9	zweispältiges	zweispältig	ADJA case=nom number=sg gender=neut degree=pos -1 6 SB

German 1:\*\* parse-text.txt All (8,99) [(Text)]



# Parsing Versions

Emacs-x86\_64-10.9@mb-mahlow: text.txt

Die Legalisierung der aktiven Sterbehilfe ist ein sehr zwiespältiges Themas. Für eine Legalisierung wur

German 1:**-	text.txt	All (1,103)	[(Text)]					
1_1	Die	der	ART	case=nom number=sg gender=fem	-1	2	NK	--
1_2	Legalisierung	Legalisierung	NN	case=nom number=sg gender=fem	-1	6	SB	--
1_3	der	der	ART	case=gen number=sg gender=fem	-1	5	NK	--
1_4	aktiven	aktiv	ADJA	case=gen number=sg gender=fem degree=pos	-1	5	NK	--
1_5	Sterbehilfe	Sterbehilfe	NN	case=gen number=sg gender=fem	-1	2	AG	--
1_6	ist	sein	VAFIN	number=sg person=3 tense=pres mood=ind	-1	0	--	--
1_7	ein	ein	ART	case=nom number=sg gender=neut	-1	10	NK	--
1_8	sehr	sehr	ADV	--	-1	9	MO	--
1_9	zwiespältiges	zwiespältig	ADJA	case=nom number=sg gender=neut degree=pos	-1	10	NK	--
1_10	Themas	Thema	NN	case=nom number=sg gender=neut	-1	6	PD	--
1_11	.	--	\$.	--	-1	10	--	--
2_1	Für	für	APPR	--	-1	4	MO	--
2_2	eine	ein	ART	case=acc number=sg gender=fem	-1	3	NK	--
2_3	Legalisierung	Legalisierung	NN	case=acc number=sg gender=fem	-1	1	NK	--
2_4	wur	wur	XY	--	-1	0	--	--

German 1:**-	parse-text.txt	All (16,94)	[(Text)]
--------------	----------------	-------------	----------



# 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