Adventures in Annotation Alignment and Error Detection

Adriane Boyd

Universität Tübingen

22 January 2018
Disclaimer

- Grateful for tools and resources
- Intend no disparagement of any particular project
- Hope that our work can help resources improve
Connection between original data and linguistic annotation is frequently not maintained in the annotation process.

**PTB Normalization and Tokenization**

*Source: Fulton Prebon (U.S.A.) Inc.*

↓

*Source: Fulton Prebon -LRB- U.S.A. -RRB- Inc.*
Adventures in Alignment

- NLP Tools: Tokenization
- User Needs: MERLIN
**Question:** How do I choose a tokenizer?

- Can I find any documentation or guidelines?
- How do I know whether a tokenizer works well with models further down the pipeline? (cf. Eckart de Castilho 2016)
- What can I do when a tokenizer doesn’t perform well?
  - Where can I find data to train a tokenizer?
  - How much data do I need?
Tokenization: Considerations

- Encoding: UTF-8 vs. ISO-8859-1
- Normalization:
  - Punctuation:
    - ‘‘Doubled’’ "ASCII" “Unicode”
    - -LRB- PTB-style parentheses -RRB-
  - Whitespace within tokens:
    - (201)_555-1234
    - out_of
- Tokenization conventions:
  - can’t → can’t
  - U.S. → U.S.

(cf. Dridan & Oepen 2012; Eckart de Castilho 2016)
Tokenization: Data for German

Very little raw data (readily) available for German:

- EmpiriST (Shared Task 2015): CMC, web data (Beißwenger et al. 2016)
  - ca. 25,000 tokens with training + test data
  - annotation guidelines

Use of detokenized data:

- Jurish & Würzner (2013): report using detokenized TIGER with some manual corrections
- de Kok (2014): report using detokenized TüBa-D/Z with OpenNLP detokenizer (rule-based)
Tokenization: Re-Aligning Tokenized Corpora

Re-Alignment

Tokenized: ( vom 18. - 20. Juni )
Original: (vom 18.-20. Juni)
Tokenization: Re-Alining Tokenized Corpora

<table>
<thead>
<tr>
<th>Re-Alignment</th>
</tr>
</thead>
</table>
| **Tokenized:** ( *vom 18. – 20. Juni*)  
**Original:** (vom 18.-20. Juni) |
Re-Alignment

Tokenized: (vom 18. - 20. Juni)
Original: (vom 18.-20. Juni)
Tokenization: Re-Aligning Tokenized Corpora

Re-Alignment

Tokenized: ( vom 18. - 20. Juni )
Original: (vom 18.-20. Juni)
Re-Alignment

Tokenized: ( vom 18. - 20. Juni )
Original:  (vom 18.-20. Juni)

TüBa-D/Z (version 10.0: 1,787,801 tokens)

- Raw texts from taz are available
- Tokenization in TüBa-D/Z treebank annotation
- Metadata links each sentence to taz article
Tokenization: TüBa-D/Z - taz Alignment

Assuming that tokenization only inserts boundaries:

- 98.2% of sentences can be aligned easily on the character level
Tokenization: TüBa-D/Z - taz Alignment

Assuming that tokenization only inserts boundaries:

- 98.2% of sentences can be aligned easily on the character level
- Remaining 1.8%?
  - Artifacts of newspaper format: ambiguous hyphenation
    Ost- Berlin → Ost-Berlin
    vs.
    An- und Abreise
    ver- spannte → ver-spannte (vs. verspannte)
  - Symbols: niklaus§taz.de → niklaus@taz.de
  - Comma quotes: (,,) → ASCII "
  - Emphasis:
    D O P P E L P O R T R A I T → DOPPELPORTRAIT
  - Other minor corrections/normalizations
Tokenization: Original vs. Detokenization

Comparing original vs detokenized texts:

- 5.5% of sentences have different tokenization

Trained tokenizer models using OpenNLP with original vs. detokenized (90/10 split):

<table>
<thead>
<tr>
<th>Test Data</th>
<th>Model</th>
<th>Orig</th>
<th>Detok</th>
</tr>
</thead>
<tbody>
<tr>
<td>Orig</td>
<td>99.92</td>
<td>99.77</td>
<td></td>
</tr>
<tr>
<td>Detok</td>
<td>99.94</td>
<td>99.95</td>
<td></td>
</tr>
</tbody>
</table>
Tokenization: Original vs. Detokenization

Comparing original vs detokenized texts:

- 5.5% of sentences have different tokenization

Trained tokenizer models using OpenNLP with original vs. detokenized (90/10 split):

<table>
<thead>
<tr>
<th>Model</th>
<th>Test Data</th>
<th>Orig</th>
<th>Detok</th>
</tr>
</thead>
<tbody>
<tr>
<td>Orig</td>
<td>99.92</td>
<td>99.77</td>
<td></td>
</tr>
<tr>
<td>Detok</td>
<td>99.94</td>
<td>99.95</td>
<td></td>
</tr>
</tbody>
</table>

- Detokenized model: many false negatives
## Tokenization: Orig. vs. Detok. Non-Whitespace

Evaluating non-whitespace tokenization:

<table>
<thead>
<tr>
<th></th>
<th>Orig</th>
<th>Detok</th>
</tr>
</thead>
<tbody>
<tr>
<td>Precision</td>
<td>99.30</td>
<td>98.93</td>
</tr>
<tr>
<td>Recall</td>
<td>99.34</td>
<td>96.65</td>
</tr>
<tr>
<td>F1</td>
<td>99.32</td>
<td>97.78</td>
</tr>
</tbody>
</table>
Tokenization: Annotated Data Uses

- Evaluate and compare tokenization approaches
- Customization of new models, e.g.:
  - without newspaper headlines
  - with custom date handling
Tokenization: Summary

Currently difficult to:

- Find tokenization annotation guidelines
- Determine how available models were trained
- Find data to evaluate and train tools

Recommendations:

- Document and publish tokenization guidelines
- Prefer annotation tools and formats that preserve alignments with original data

Would like to have a tokenizer evaluation tool that:

- Compares a set of available tokenizers on a test corpus
- Shows learning curves for tokenizer training
Tokenization: Summary

Currently difficult to:

- Find tokenization annotation guidelines
- Determine how available models were trained
- Find data to evaluate and train tools

Recommendations:

- Document and publish tokenization guidelines
- Prefer annotation tools and formats that preserve alignments with original data
Tokenization: Summary

Currently difficult to:

- Find tokenization annotation guidelines
- Determine how available models were trained
- Find data to evaluate and train tools

Recommendations:

- Document and publish tokenization guidelines
- Prefer annotation tools and formats that preserve alignments with original data

Would like to have a tokenizer evaluation tool that:

- Compares a set of available tokenizers on a test corpus
- Shows learning curves for tokenizer training
Adventures in Annotation Alignment and Error Detection
Adriane Boyd

Introduction
Alignment
Tokenization
MERLIN
Error Detection
Introduction
DECCA Project
POS
Treebank
Spans
Further Work
Conclusion
References

MERLIN
Illustrates CEFR scales levels in a written learner corpus for Czech, German, and Italian in a didactically-motivated online platform

The MERLIN corpus
MERLIN provides access to 2,256 texts written by learners of Czech, Italian and German.
The learner texts stem from standardized language tests and they have been reliably related to the CEFR levels.

Use MERLIN ...
... to better understand the levels of the Common European Framework of Reference (CEFR).

Video tutorial ...

The MERLIN corpus

Simple search
Advanced search
Define a subcorpus
Statistics

Search for words in the learner texts and display them in context (e.g., full text):
examples: watchin, grip, rid

Search in: learner text
Corpus: entire collection

No subcorpus yet? Compile your own text collection here.

Home | Search | About MERLIN | ?
Czech | German | English | Italian
Adventures in Annotation
Alignment and Error Detection
Adriane Boyd

Introduction
Alignment
Tokenization
MERLIN
Error Detection
Introduction
DECCA Project
POS
Treebank
Spans
Further Work
Conclusion
References

MERLIN Team

University of Technology Dresden (coordination)
  Katrin Wisniewski, Maria Lieber, Claudia Woldt, Karin Schöne

European Academy Bozen
  Andrea Abel, Verena Blaschitz, Verena Lyding, Lionel Nicolas, Chiara Vettori

Charles University Prague
  Kateřina Vodičková, Pavel Pečený, Jirka Hana, Veronika Čurdová

telc Frankfurt/Main
  Sybille Plassmann, Gudrun Klein, Louise Lauppe

Berufsförderungsinstitut Oberösterreich, Linz
  Gerhard Zahrer, Pia Zaller

Eberhard Karls University Tübingen
  Detmar Meurers, Adriane Boyd, Serhiy Bykh, Julia Krivanek
MERLIN Corpus

- Approx. 200 texts per CEFR level
  - Czech (A2–B2): 441 texts
  - German (A1–C1): 1033 texts
  - Italian (A1–B2): 813 texts

- Detailed re-ratings:
  - overall
  - orthography
  - grammatical accuracy
  - vocabulary range
  - vocabulary control
  - coherence & cohesion
  - sociolinguistic appropriateness

- Learner metadata
- Task descriptions
MERLIN Annotations

Manual

- transcription: task citations, greetings, closings, ...
- target hypotheses (normalization)
- error annotation

Automatic

- tokens, sentences
- lemmas, POS
- dependency parses
- repetitions within texts

Derived

- statistical measures for error annotation
e.g., word order errors per token
MERLIN Platform

- Target audiences
  - language teachers
  - test and curriculum developers
  - textbook authors
  - (computational) linguists

- Search engines
  - simple (solr): KWIC, formatted full texts, metadata
  - advanced (ANNIS): full TH/EA, automatic annotations, metadata
Key word in context
Kind krank und es bett bleiben muss Darum Ich

learner text:
Liebe Julia, 12.03.2012
Alles Liebe
deine
Eva

target hypothesis 1:
Alles Liebe deine Eva
## Introduction

**Alignment**
- **Tokenization**
- **MERLIN**
- **Error Detection**

### Alignment

#### Tokenization

**MERLIN**

#### Error Detection

**DECCA Project**

**POS**

**Treebank**

**Spans**

**Further Work**

### Conclusion

### References

### MERLIN: Advanced Search Results

<table>
<thead>
<tr>
<th>TH/EA (grid)</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Meine</td>
<td>Kind</td>
</tr>
<tr>
<td></td>
<td>krank</td>
</tr>
<tr>
<td>Mein</td>
<td>Kind</td>
</tr>
<tr>
<td>CHA</td>
<td>INS</td>
</tr>
<tr>
<td>Mein</td>
<td>Kind</td>
</tr>
<tr>
<td>CHA</td>
<td>INS</td>
</tr>
<tr>
<td>G_Verb_compl</td>
<td>G_Verb_compl</td>
</tr>
<tr>
<td>G_Morphol_Wrong</td>
<td>G_Vo</td>
</tr>
<tr>
<td>ambig</td>
<td>O_Punct</td>
</tr>
<tr>
<td></td>
<td>O_Capit</td>
</tr>
<tr>
<td>o</td>
<td>pos</td>
</tr>
<tr>
<td>womaincl</td>
<td>o</td>
</tr>
</tbody>
</table>
## MERLIN: Annotation Pipeline

<table>
<thead>
<tr>
<th>Format</th>
<th>Tool</th>
<th>Annotation</th>
</tr>
</thead>
<tbody>
<tr>
<td>hand-written scan</td>
<td></td>
<td></td>
</tr>
<tr>
<td>custom XML</td>
<td>XMLmind</td>
<td>transcription</td>
</tr>
<tr>
<td>PAULA</td>
<td>custom converter</td>
<td>tokens, sentences</td>
</tr>
<tr>
<td>Exmaralda XML</td>
<td>SaltNPepper</td>
<td>TH1</td>
</tr>
<tr>
<td></td>
<td>Exmaralda/Falko</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Excel add-ins</td>
<td></td>
</tr>
<tr>
<td>PAULA</td>
<td>custom converter</td>
<td></td>
</tr>
<tr>
<td>MMAX2</td>
<td>SaltNPepper</td>
<td>EA1</td>
</tr>
<tr>
<td>PAULA</td>
<td>SaltNPepper</td>
<td></td>
</tr>
</tbody>
</table>
## MERLIN: Annotation Pipeline, cont.

<table>
<thead>
<tr>
<th>Format</th>
<th>Tool</th>
<th>Annotation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exmaralda XML</td>
<td>SaltNPepper (improved)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Exmaralda/Falko</td>
<td>TH2</td>
</tr>
<tr>
<td></td>
<td>Excel add-ins</td>
<td></td>
</tr>
<tr>
<td>PAULA</td>
<td>custom converter</td>
<td></td>
</tr>
<tr>
<td>MMAX2</td>
<td>SaltNPepper</td>
<td>EA2</td>
</tr>
<tr>
<td>PAULA</td>
<td>SaltNPepper</td>
<td></td>
</tr>
<tr>
<td>PAULA</td>
<td>custom UIMA pipeline</td>
<td>automatic</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>→ solr</td>
<td>custom converter</td>
<td></td>
</tr>
<tr>
<td>→ ANNIS</td>
<td>SaltNPepper</td>
<td></td>
</tr>
</tbody>
</table>
MERLIN: Problematic Conversions

- PAULA → Exmaralda XML (SaltNPepper)
  - Tokens only, whitespace/formatting is lost

Liebe Julia, 12.03.2012
wie geht´s? Mir geht es gut. Ich habe eine Fahrkarte für einen

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>tok</td>
<td>ctk</td>
</tr>
<tr>
<td>2</td>
<td>Liebe</td>
<td>Liebe</td>
</tr>
<tr>
<td>3</td>
<td>Julia</td>
<td>Julia</td>
</tr>
<tr>
<td>4</td>
<td>,</td>
<td>,</td>
</tr>
<tr>
<td>5</td>
<td>12.03.2012</td>
<td>12.03.2012</td>
</tr>
<tr>
<td>6</td>
<td>wie</td>
<td>wie</td>
</tr>
<tr>
<td>7</td>
<td>geht</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>’</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>s</td>
<td>geht´s</td>
</tr>
</tbody>
</table>
MERLIN: Complicating Factors

One annotation step uses Excel, where annotators can (and do!) edit almost anything

- **Advantage:** annotators can potentially edit transcription or annotations
- **Disadvantage:** annotator can accidentally edit transcription or annotations

⇒ Re-aligning raw data and annotation is complicated
MERLIN: Cascading Errors

- MERLIN tokenization guidelines: *geht´s* is one token
- Cascading errors in pipeline of standard German NLP tools:
  - STTS tagset has no tag for this contraction
MERLIN: Cascading Errors

- MERLIN tokenization guidelines: \textit{geht´s} is one token
- Cascading errors in pipeline of standard German NLP tools:
  - STTS tagset has no tag for this contraction

<table>
<thead>
<tr>
<th>learner</th>
<th>wie</th>
<th>geht´z</th>
<th>dir</th>
<th>und</th>
<th>wie</th>
<th>geht´z</th>
<th>deine</th>
<th>Eltern</th>
</tr>
</thead>
<tbody>
<tr>
<td>tok_lemma</td>
<td>wie</td>
<td>geht´z</td>
<td>du</td>
<td>und</td>
<td>wie</td>
<td>geht´z</td>
<td>dein</td>
<td>Eltern</td>
</tr>
<tr>
<td>tok_lemma_bohnet</td>
<td>wie</td>
<td>geht´z</td>
<td>dir</td>
<td>und</td>
<td>wie</td>
<td>geht´z</td>
<td>dein</td>
<td>Eltern</td>
</tr>
<tr>
<td>tok_pos</td>
<td>KOUS</td>
<td>VVFIN</td>
<td>PRF</td>
<td>KON</td>
<td>KOUS</td>
<td>VVFIN</td>
<td>PPOSAT</td>
<td>NN</td>
</tr>
<tr>
<td>tok_pos_bohnet</td>
<td>PWAV</td>
<td>VVFIN</td>
<td>PPER</td>
<td>KON</td>
<td>KOKOM</td>
<td>ADV</td>
<td>PPOSAT</td>
<td>NN</td>
</tr>
<tr>
<td>tok_pos_stanford</td>
<td>PWAV</td>
<td>VVFIN</td>
<td>PRF</td>
<td>KON</td>
<td>KOKOM</td>
<td>NN</td>
<td>PPOSAT</td>
<td>NN</td>
</tr>
</tbody>
</table>
MERLIN: Summary

- User needs may require annotations to be aligned with original formatted texts
- Maintaining whitespace / formatting in a long pipeline is difficult
  - Every single tool needs to support, e.g., character offsets
Alignment between raw data and annotation is crucial for certain tools and use cases:

**Aligned Tokens**

Tokenized: (vom 18. - 20. Juni)
Original: (vom 18.-20. Juni)
Alignment between raw data and annotation is crucial for certain tools and use cases:

**Aligned Tokens**

<table>
<thead>
<tr>
<th>Tokenized:</th>
<th>( vom 18. – 20. Juni )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original:</td>
<td>(vom 18.-20. Juni)</td>
</tr>
</tbody>
</table>
From Alignment to Error Detection

Alignment between raw data and annotation is crucial for certain tools and use cases:

**Aligned Tokens**

Tokenized: (vom 18. – 20. Juni)
Original: (vom 18.-20. Juni)
From Alignment to Error Detection

Alignment between raw data and annotation is crucial for certain tools and use cases:

Aligned Tokens

Tokenized: ( vom 18. – 20. Juni )
Original: (vom 18.-20. Juni)
From Alignment to Error Detection

Alignment between raw data and annotation is crucial for certain tools and use cases:

**Aligned Tokens**

Tokenized: ( vom 18. – 20. Juni )
Original: (vom 18.-20. Juni)

What about the quality of the annotation itself?

**Annotated Tokens**

Tokenized: ( vom 18. – 20. Juni )
POS Tags: $( APPRART ADJA APPR ADJA NN $( KON
Error Detection: Introduction

Annotated corpora are used:

- To train and test NLP technology
- For searching for linguistically relevant patterns
Error Detection: Introduction

Annotated corpora are used:

- To train and test NLP technology
- For searching for linguistically relevant patterns

Improving corpus annotation:

- More reliable training and evaluation
- Higher precision and recall in corpus searches
Error Detection: Introduction

Automatic annotation error detection: find inconsistencies in corpus annotation with respect to

- Internal: statistical model based on data within corpus
- External: grammatical model, other external resource

Good overview in Dickinson (2015)

Co-PIs: Markus Dickinson and Detmar Meurers

Website: http://decca.osu.edu

Methods for automatic error detection and correction in:

- Part-of-speech tags (Dickinson & Meurers 2003a)
- Treebanks (Dickinson & Meurers 2003b, 2005c)
- Discontinuous treebanks (Dickinson & Meurers 2005b; Dickinson 2005)
- Spoken language corpora (Dickinson & Meurers 2005a)
- Dependencies (Boyd et al. 2008)
- And related issues (Boyd et al. 2007a,b)
Error Detection: DECCA
Detection of Errors and Correction in Corpus Annotation

**Variation** \( n \)-gram method: identify repeated material in a corpus that appears with different annotations.
Error Detection: DECCA
Detection of Errors and Correction in Corpus Annotation

**Variation** *n*-gram method: identify repeated material in a corpus that appears with different annotations

Variation can result from:
- genuine ambiguity
- inconsistent annotation
Error Detection: DECCA
Detection of Errors and Correction in Corpus Annotation

Variation *n*-gram method: identify repeated material in a corpus that appears with different annotations

Variation can result from:

▶ genuine ambiguity
▶ inconsistent annotation

WSJ POS Annotation

<table>
<thead>
<tr>
<th>n’t/RB</th>
<th>elaborate/VB</th>
</tr>
</thead>
<tbody>
<tr>
<td>n’t/RB</td>
<td>elaborate/VB</td>
</tr>
<tr>
<td>n’t/RB</td>
<td>elaborate/JJ</td>
</tr>
</tbody>
</table>
Error Detection: DECCA
Detection of Errors and Correction in Corpus Annotation

Variation \textit{n-gram method}: identify repeated material in a corpus that appears with different annotations

Variation can result from:

- genuine ambiguity
- inconsistent annotation

**WSJ POS Annotation**

<table>
<thead>
<tr>
<th></th>
<th>would</th>
<th>n’t/RB</th>
<th>elaborate/VB</th>
</tr>
</thead>
<tbody>
<tr>
<td>6:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2:</td>
<td>did</td>
<td>n’t/RB</td>
<td>elaborate/VB</td>
</tr>
<tr>
<td>1:</td>
<td>did</td>
<td>n’t/RB</td>
<td>\textcolor{red}{elaborate/JJ}</td>
</tr>
</tbody>
</table>
Error Detection: DECCA Algorithm

Extract all $n$-grams containing a token that is annotated differently in another occurrence of the $n$-gram in the corpus.

- **variation nucleus**: recurring unit with different annotation
- **variation $n$-gram**: variation nucleus with identical context
Error Detection: DECCA Algorithm

Extract all $n$-grams containing a token that is annotated differently in another occurrence of the $n$-gram in the corpus.

- variation nucleus: recurring unit with different annotation
- variation $n$-gram: variation nucleus with identical context

To be efficient, algorithm calculates variation $n$-grams based on variation ($n - 1$)-grams

- Instance of Apriori algorithm (Agrawal & Srikant 1994)
Error Detection: POS Annotation
Dickinson & Meurers (2003a)

WSJ POS Annotation

- LRB- During its centennial year, The Wall Street Journal will report events of the past century that T stand as milestones of American business history. -RRB-

- 5 times as DT (determiner)
- 5 times as WDT (wh-determiner)

How to determine whether we have ambiguity or an error?

- Context: the more similar the surrounding context, the higher the likelihood of an error
Error Detection: Heuristics
Dickinson & Meurers (2003a)

To improve precision:

- Longer variation $n$-grams are more likely to be errors

that (DT vs. IN vs. RB vs. WDT)

that/??
Error Detection: Heuristics
Dickinson & Meurers (2003a)

To improve precision:

▷ Longer variation *n*-grams are more likely to be errors

that (DT vs. IN vs. RB vs. WDT)

events of the past century that/WDT *T* stand as
Error Detection: Heuristics

Dickinson & Meurers (2003a)

To improve precision:

- Longer variation $n$-grams are more likely to be errors

  that (DT vs. IN vs. RB vs. WDT)

  events of the past century that/WDT *T* stand as

- Distrust the fringe

  decided (VBD vs. VBN)

  decided/VB? how it will
  decided/ VB? how it will
Error Detection: Heuristics
Dickinson & Meurers (2003a)

To improve precision:

▶ Longer variation $n$-grams are more likely to be errors

that (DT vs. IN vs. RB vs. WDT)

events of the past century that/WDT *T* stand as

▶ Distrust the fringe

decided (VBD vs. VBN)

he has decided/VBN how it will
he decided/VBD how it will
Error Detection: WSJ POS Results
Dickinson & Meurers (2003a)

WSJ corpus:
- 1,289,201 tokens
- 98.2% appear more than once

Sampling 7,141 distinct non-fringe variation \( n \)-gram types for \( 3 \leq n \leq 224 \):
- 92.8% are errors \( \rightarrow \) each at least one correction
- Given 3% estimated POS error rate in the WSJ, the method has a POS error recall of at least 17%
Error Detection: Treebank Annotation
Dickinson & Meurers (2003b)

WSJ Treebank Annotation

many of whom *T*
Error Detection: Treebank Annotation
Dickinson & Meurers (2003b)

WSJ Treebank Annotation

many of whom *T*

- Different tags: WHPP vs. PP
Error Detection: Treebank Annotation
Dickinson & Meurers (2003b)

WSJ Treebank Annotation

all of whom *T*

<Diagram>
Error Detection: Treebank Annotation
Dickinson & Meurers (2003b)

WSJ Treebank Annotation

all of whom *T*

- Annotated vs. not: WHPP vs. NIL
### Error Detection: Treebank Output

Sample WSJ 3-gram output:

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>to say</td>
<td>VP</td>
<td>good 0 * to say <em>T</em> about the pound 's</td>
</tr>
<tr>
<td><em>T</em></td>
<td>NIL</td>
<td>good 0 * to say <em>T</em> about the pound 's</td>
</tr>
<tr>
<td>an example of</td>
<td>NP</td>
<td>as an example of the</td>
</tr>
<tr>
<td>an example of</td>
<td>NIL</td>
<td>as an example of the</td>
</tr>
<tr>
<td>year to year</td>
<td>NIL</td>
<td>from year to year .</td>
</tr>
<tr>
<td>year to year</td>
<td>NP</td>
<td>from year to year .</td>
</tr>
</tbody>
</table>

- Algorithm is run separately for each possible constituent length 1..n
DECCA treebank algorithm can be applied to any continuous span annotation.

Examples:

- Named entities (TüBa-D/Z 10.0)
- Error annotation (EFCAMDAT2)
## Error Detection: Named Entities in TüBa-D/Z

<table>
<thead>
<tr>
<th>NIL</th>
<th>&quot; Roten Rad &quot;</th>
</tr>
</thead>
<tbody>
<tr>
<td>NX=OTH</td>
<td>&quot; Roten Rad &quot;</td>
</tr>
<tr>
<td>NX=PER</td>
<td>der Befreiungsfront <em>Farabundo Marti</em> ( FMLN )</td>
</tr>
<tr>
<td>NX=PER</td>
<td>Befreiungsfront <em>Farabundo Marti</em> ( FMLN )</td>
</tr>
<tr>
<td>NX=PER</td>
<td>der Befreiungsfront <em>Farabundo Marti</em> ( FMLN )</td>
</tr>
<tr>
<td>NX=ORG</td>
<td>Befreiungsfront <em>Farabundo Marti</em> ( FMLN )</td>
</tr>
<tr>
<td>SIMPX=OTH</td>
<td>&quot; Indien verstehen &quot;</td>
</tr>
<tr>
<td>SIMPX=ORG</td>
<td>&quot; Indien verstehen &quot;</td>
</tr>
<tr>
<td>NX=PER</td>
<td>&quot; Forrest Gump &quot;</td>
</tr>
<tr>
<td>NX=OTH</td>
<td>&quot; Forrest Gump &quot;</td>
</tr>
</tbody>
</table>
EFCAMDAT2 (Geertzen et al. 2013): English L2 learner corpus with 83 million words from 1 million assignments written by 174,000 learners (A1 – C2)

- Partially annotated with feedback provided by language teachers

Text with Feedback

I’m from Brazil, So Paulo {XC: So Paulo, in Brazil}

... I’m married and my wife is twenty-eighty {SP: eight} .

... Glad to meet you __ {PU:} !
Error Detection: EFCAMDAT Output

<table>
<thead>
<tr>
<th>Category</th>
<th>Fringe</th>
<th>Correction</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>NIL</td>
<td>0</td>
<td>--</td>
<td>years old. I’m from Brazil, So Paulo. There</td>
</tr>
<tr>
<td>NIL</td>
<td>0</td>
<td>--</td>
<td>. I’m from Brazil, So Paulo. There are</td>
</tr>
<tr>
<td>NIL</td>
<td>0</td>
<td>--</td>
<td>old. I’m from Brazil, So Paulo. There</td>
</tr>
<tr>
<td>NIL</td>
<td>0</td>
<td>--</td>
<td>years old. I’m from Brazil, So Paulo.</td>
</tr>
<tr>
<td>NIL</td>
<td>0</td>
<td>--</td>
<td>years old. I’m from Brazil, So Paulo.</td>
</tr>
<tr>
<td>NIL</td>
<td>0</td>
<td>--</td>
<td>. I’m from Brazil, So Paulo. There</td>
</tr>
<tr>
<td>NIL</td>
<td>0</td>
<td>--</td>
<td>I’m from Brazil, So Paulo. There are</td>
</tr>
<tr>
<td>NIL</td>
<td>0</td>
<td>--</td>
<td>from Brazil, So Paulo, but I live in</td>
</tr>
<tr>
<td>WO</td>
<td>0</td>
<td>Sao Paulo, Brazil</td>
<td>in Brazil, So Paulo.</td>
</tr>
<tr>
<td>XC</td>
<td>0</td>
<td>So Paulo, in Brazil</td>
<td>years old. I’m from Brazil, So Paulo. There are</td>
</tr>
<tr>
<td>WO</td>
<td>0</td>
<td>So Paulo, Brazil</td>
<td>years old. I’m from Brazil, So Paulo.</td>
</tr>
<tr>
<td>WO</td>
<td>0</td>
<td>So Paulo, Brazil</td>
<td>from Brazil, So Paulo, but I live in</td>
</tr>
</tbody>
</table>

Another instance:

<table>
<thead>
<tr>
<th>Category</th>
<th>Correction</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>D / MW</td>
<td>– / Brazil</td>
<td>I’m from Brazil, So Paulo</td>
</tr>
</tbody>
</table>

→ DECCA can be used to explore/evaluate crowd-sourced annotations
Error Detection: Increasing Recall
Boyd et al. (2007a)

Two ways to increase recall:

- Redefine *variation nuclei* to extend the set of what counts as recurring data

  **Variation Nuclei**
  
  many *of whom* → many *of {which/whom}*

- Redefine *context* and *heuristics* to obtain more variation *n*-grams

  **Context**
  
  many *of whom* → {some/many/most/all}/DT *of whom*
Error Detection: Discontinuous Spans
Dickinson & Meurers (2005b)

From TIGER Treebank

in diesem Punkt seien **sich** Bonn und London **nicht** **einig**

vs.

in diesem Punkt seien **sich** Bonn und London **offensichtlich nicht** **einig**

‘Bonn and London (clearly) do not agree on this point’

- AP vs. NIL
Error Detection: Dependencies

Boyd et al. (2008)

From TigerDB (Forst et al. 2004):

▶ SB vs. NIL
Error Detection: Summary

Automatic error detection:

- Leads to improved corpus quality for NLP / search
- Provides feedback to corpus developers for annotation scheme design and documentation

DECCA variation \(n\)-gram approach:

- Finds errors in token, span, discontinuous span, and dependency annotation
- Does not depend on language, corpus, or tagset

Website: http://decca.osu.edu

Download DECCA code:

http://github.com/adrianeboyd/decca
Conclusion

Corpus annotation with

▶ explicit links to the original data
▶ attention to consistency through error detection, etc.
  ▶ which informs annotation guidelines

has a wider range of

▶ potential users
  ▶ non-specialists (e.g., language teachers)
▶ potential uses
  ▶ gold standard in evaluations
  ▶ high-quality, customizable training data
References


Adventures in Annotation
Alignment and Error Detection
Adriane Boyd

Introduction

Alignment

Tokenization

MERLIN

Error Detection

Introduction

DECCA Project

POS

Treebank

Spans

Further Work

Conclusion

References


http://aclweb.org/anthology/P05-1040.
Adventures in Annotation
Alignment and Error Detection
Adriane Boyd

Introduction
Alignment
Tokenization
MERLIN
Error Detection
Introduction
DECCA Project
POS
Treebank
Spans
Further Work
Conclusion
References


Adventures in Annotation Alignment and Error Detection
Adriane Boyd

Introduction
Alignment
Tokenization
MERLIN
Error Detection
DECCA Project
POS
Treebank
Spans
Further Work
Conclusion
References


