Text Corpus Format

(Version 0.4)

Helmut Schmid
SfS Tübingen / IMS Stuttgart

Why Do We Need a TCF?

- We want to chain together many different NLP tools (tokenizers, POS taggers, lemmatizers, parsers etc.)

Why Do We Need a TCF?

- We want to chain together many different NLP tools (tokenizers, POS taggers, lemmatizers, parsers etc.)
- Typically, each NLP tool uses its own idiosyncratic input/output format.

Why Do We Need a TCF?

- We want to chain together many different NLP tools (tokenizers, POS taggers, lemmatizers, parsers etc.)
- Typically, each NLP tool uses its own idiosyncratic input/output format.
- Writing input/output converters for each possible combination of tools is too expensive. (5 tokenizers x 5 POS taggers = 25 converters)

Why Do We Need a TCF?

→ Usage of a common data format

→ Wrappers turn NLP tools into webservices which use the common data format

Why Define a New Format?

Existing data formats (such as TEI, MAF, Tiger, ...)

- do not cover all we need
  (e.g. coreference links, named entitities, etc.).
- have not been developed for webservice applications
  (too detailed, too inefficient, too difficult to process, ...)
- are difficult to extend
  (Committees need to to be convinced)

General Priciples of TCF

- uses an XML encoding
- only supports UTF-8
- uses stand-off annotation
  - Each annotation (tokens, sentences, POS tags, lemmas, parse trees) is stored in a separate **layer**
  - All annotations are stored in a single XML file

A Simple Text File in TCF

```
<D-Spin version="0.4">
    <TextCorpus lang="en">
        <text>This is a sentence. That‘s another one.</text>
    </TextCorpus>
</D-Spin>
```

A Simple Text File in TCF

```
<D-Spin version="0.4">
    <TextCorpus lang="en">
        <text>This is a sentence. That‘s another one.</text>
    </TextCorpus>
</D-Spin>
```

TCF Version number

A Simple Text File in TCF

```
<D-Spin version="0.4">
    <TextCorpus lang="en">
        <text>This is a sentence. That‘s another one.</text>
    </TextCorpus>
</D-Spin>
```

Language attribute

A Simple Text File in TCF

```
<D-Spin version="0.4">
    <TextCorpus lang="en">
        <text>This is a sentence. That‘s another one.</text>
    </TextCorpus>
</D-Spin>
```

Actual text is surrounded by a <text> element

A Tokenized Text File

```
<D-Spin version="0.4">
    <TextCorpus lang="en">
        <text>This is a sentence. That‘s another one.</text>
        <tokens>
            <token>This</token>
            <token>is</token>
            <token>a</token>
            <token>sentence</token>
            <token>.</token>
            <token>That</token>
            <token>‘s</token>
            <token>another</token>
            <token>one</token>
            <token>.</token>
        </tokens>
    </TextCorpus>
</D-Spin>
```

Tokens are surrounded by a <token> element

A Tokenized Text File

```
<D-Spin version="0.4">
    <TextCorpus lang="en">
        <text>This is a sentence. That‘s another one.</text>
        <tokens>
            <token>This</token>
            <token>is</token>
            <token>a</token>
            <token>sentence</token>
            <token>.</token>
            <token>That</token>
            <token>‘s</token>
            <token>another</token>
            <token>one</token>
            <token>.</token>
        </tokens>
    </TextCorpus>
</D-Spin>
```

All tokens are surrounded by a <tokens> element which forms an annotation layer

A Tokenized Text File

```
<D-Spin version="0.4">
    <TextCorpus lang="en">
        <text>This is a sentence. That‘s another one.</text>
        <tokens>
            <token ID="t1">This</token>
            <token ID="t2">is</token>
            <token ID="t3">a</token>
            <token ID="t4">sentence</token>
            <token ID="t5">.</token>
            <token ID="t6">That</token>
            <token ID="t7">‘s</token>
            <token ID="t8">another</token>
            <token ID="t9">one</token>
            <token ID="t10">.</token>
        </tokens>
    </TextCorpus>
</D-Spin>
```

optional reference IDs

Sentence Boundaries

```
<TextCorpus lang="en">
    <tokens>
        <token ID="t1">This</token>
        <token ID="t2">is</token>
        <token ID="t3">a</token>
        <token ID="t4">sentence</token>
        <token ID="t5">.</token>
        <token ID="t6">That</token>
        <token ID="t7">‘s</token>
        <token ID="t8">another</token>
        <token ID="t9">one</token>
        <token ID="t10">.</token>
    </tokens>
    <sentences>
        <sentence tokenIDs="t1 t2 t3 t4 t5"/>
        <sentence tokenIDs="t6 t7 t8 t9 t10"/>
    </sentences>
</TextCorpus>
```

sentences are identified by a list of token references
(new in version 0.4)

Part-of-Speech Tags

```
<TextCorpus lang="en">
    <tokens>
        <token ID="t1">This</token>
        <token ID="t2">is</token>
        <token ID="t3">a</token>
        <token ID="t4">sentence</token>
        ...
    </tokens>
    <POStags  tagset="PennTB">
        <tag  tokenIDs="t1">DT</tag>
        <tag  tokenIDs="t2">VBZ</tag>
        <tag  tokenIDs="t3">DT</tag>
        <tag  tokenIDs="t4">NN</tag>
        ...
    </POStags>
</TextCorpus>
```

POStags layer

- tagset attribute

Part-of-Speech Tags

```
<TextCorpus lang="en">
    <tokens>
        <token ID="t1">This</token>
        <token ID="t2">is</token>
        <token ID="t3">a</token>
        <token ID="t4">sentence</token>
        ...
    </tokens>
    <POStags  tagset="PennTB">
        <tag  tokenIDs="t1">DT</tag>
        <tag  tokenIDs="t2">VBZ</tag>
        <tag  tokenIDs="t3">DT</tag>
        <tag  tokenIDs="t4">NN</tag>
        ...
    </POStags>
</TextCorpus>
```

POStags layer

- tag element
- list of token IDs (→ MAF)

Lemma Annotations

```
<TextCorpus lang="en">
    <tokens>
        <token ID="t1">This</token>
        <token ID="t2">is</token>
        ...
    </tokens>
    <POStags tagset="PennTB">
        <tag tokenIDs="t1">DT</tag>
        ...
    </POStags>
    <lemmas>
        <lemma tokenIDs="t1">this</lemma>
        <lemma tokenIDs="t2">be</lemma>
        ...
    </lemmas>
</TextCorpus>
```

lemmas layer

- analogous to POStags

Constituent Parse Trees

```
<TextCorpus lang="en">
    <tokens>
        <token ID="t1">This</token>
        ...
    </tokens>
    <parsing tagset="PennTB">
        <parse>
            <constituent ID="c1" cat= "S">
                <constituent ID="c2" cat= "NP">
                    <constituent ID="c3" cat= "DT" tokenIDs="t1"/>
                </constituent>
                ...
        </parse>
    </parsing>
</TextCorpus>
```

parsing layer

- tagset attribute

Constituent Parse Trees

```
<TextCorpus lang="en">
    <tokens>
        <token ID="t1">This</token>
        ...
    </tokens>
     <parsing tagset="PennTB">
        <parse>
            <constituent ID="c1" cat= "S">
                <constituent ID="c2" cat= "NP">
                    <constituent ID="c3" cat= "DT" tokenIDs="t1"/>
                </constituent>
                ...
        </parse>
    </parsing>
</TextCorpus>
```

parsing layer

- nested constituents
- token references

Dependency Parse Trees

```
<TextCorpus lang="en">
    <tokens>
        <token ID="t1">This</token>
        ...
    </tokens>
    <depparsing tagset="dep">
        <parse>
            <dependency func="SUBJ" depIDs="t1" govIDs="t2"/>
            <dependency func="SPEC" depIDs="t3" govIDs="t4"/>
        </parse>
    </depparsing>
</TextCorpus>
```

depparsing layer

Dependency Parse Trees

```
<TextCorpus lang="en">
    <tokens>
        <token ID="t1">This</token>
        ...
    </tokens>
    <depparsing tagset="dep">
        <parse>
            <dependency func="SUBJ" depIDs="t1" govIDs="t2"/>
            <dependency func="SPEC" depIDs="t3" govIDs="t4"/>
        </parse>
    </depparsing>
</TextCorpus>
```

depparsing layer

- flat list of dependencies
- references to tokens or POS tags

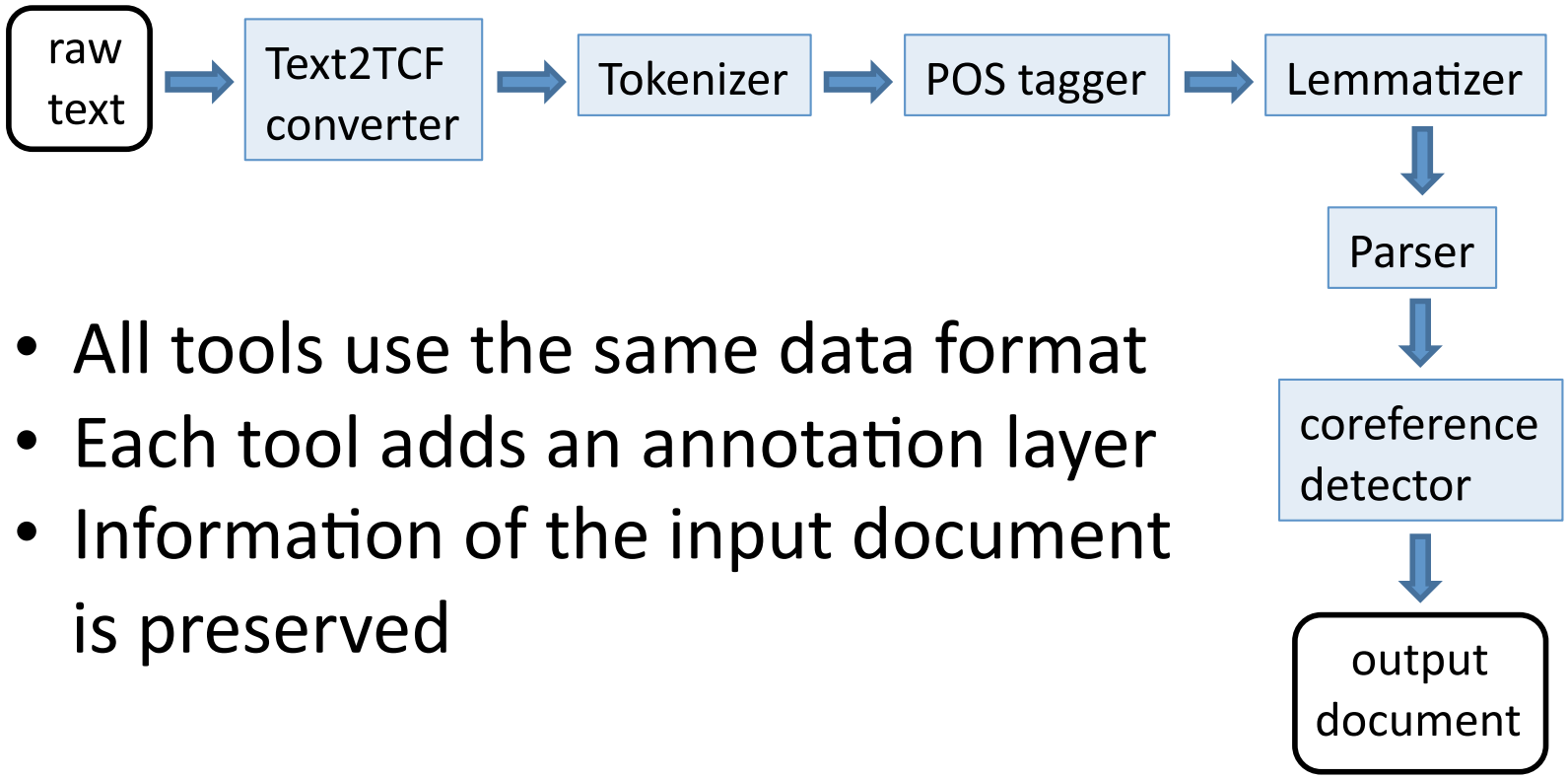
Relations

```
<TextCorpus lang="en">
    <tokens>
        <token ID="t1">This</token>
        ...
    </tokens>
    <relations type="coference">
        <relation refIDs="c3 c17"/>
        <relation refIDs="c7 c13"/>
    </relations>
</TextCorpus>
```

relation layer

- here used to represent constituent coreferences

Tool Chaining



- All tools use the same data format
- Each tool adds an annotation layer
- Information of the input document is preserved

Wrapper

- Software which implements
  - the conversion between tool-specific data formats and TCF
  - the webservice functionality
- internally calls the original tool
- A separate wrapper needed for each tool
- No modification of the original tool required

The End