Appendix

A Schema for Augmented Text

This schema defines the elements and attributes that are added to a document during linguistic preprocessing (cf. Sec. 12.1).

```
# A RELAX NG compact syntax pattern for text augmented by
# linguistic preprocessing

# The namespace used:
namespace aug = "http://www.purl.org/net/ties/schema/augment"

# The start element. Augmented text contains any number of suitable subelements.
# It can be embedded within any XML elements from outer namespaces.
start = ( AugmentedText |OtherOutsideContent )
AugmentedText = element aug:augment  { TopLevelContent* }


# The content (allowed subelements) of elements:

# At the outmost level, any elements are allowed (the sentence element is
# optional).
TopLevelContent = ( Sentence |InlineContent |OtherContent )

# Any elements except sentences are allowed as inline content. Sentences can
# directly contain POS elements, e.g. punctuation; constituents can recursively
# contain themselves.
InlineContent = ( Constituent |POS |OtherContent )


# Other Elements from this namespace (in alphabetic order):

# A sentence constituent:
Constituent = element aug:const {
  TypeAttribute,     # required attribute
  InlineContent+     # must contain one or more suitable subelements
}

# A part-of-speech (word or other token):
POS = element aug:pos {
  TypeAttribute,                              # required attribute
```

```
  NormalAttribute?, SegmentationAttributes?, # optional attributes
  text                                       # the actual word or token
}

# A sentence contains one or more suitable subelements:
Sentence = element aug:sent { InlineContent+ }


# Elements from other namespaces:
# Any elements from other namespaces are allowed, as long as they embed the
# elements from this schema in the appropriate way. This allows embedding
# augmented text in any kind of XML documents.
# No other elements are allowed in POS (which contains only a text token).

# Outside: can contain any mixed contents and our start element as well as
# top-level elements (so the start element is optional for embedded augmented
# text).
OtherOutsideContent = element * - aug:* {
  AnyAttributes,
  ( text |OtherOutsideContent |AugmentedText |TopLevelContent )*
}

# Any elements from other namespaces are allowed, as long as they embed the
# elements from this schema in the appropriate way. Sentences and constituents
# cannot directly contain other sentences, but embedded foreign elements can
# (e.g., footnotes).
OtherContent = element * - aug:* { AnyAttributes, TopLevelContent* }

# Other elements can contain any number of attributes
AnyAttributes = ( attribute * { text } )*


# Attributes (in alphabetic order):

# The normalized form of an element (when different from the textual content).
# Can contain pipe-separated alternatives, e.g.:
#    <pos type="PRF" normal="er|es|sie|Sie">sich</pos>
NormalAttribute = attribute normal { text }

# Compound segmentation (relevant for German texts).
# Example: <pos type="NE" normal="Rettungsroboter"
#       segments="rettung s roboter" normalSegments="rettung roboter"
#       baseSegment="roboter">Rettungsroboter</pos>
# Normalized forms can contain pipe-separated alternatives, e.g.:
```

```
#     <pos type="NN" segments="wettbewerbs aufgaben"
#        normalSegments="wettbewerb aufgeb|aufgabe" baseSegment="aufgeb|aufgabe"
#        >Wettbewerbsaufgaben</pos>
SegmentationAttributes = {
  # Whitespace-separated list of segments
  attribute segments { list { text+ } },
  # Whitespace-separated list of the normalized form of segments (when known)
  attribute normalSegments { list { text+ } },
  # The normalized form of the main segment
  attribute baseSegment { text }
}

# The type of an element. The value "other" indicates an element that
# could not be classified (so no mixed content is required).
TypeAttribute = attribute type { xsd:NMTOKEN }

# Not part of the schema (preprocessor/language-dependent): Enumerations
# of attribute values, e.g. attribute type { "nc"|"vc"|"pc" }.
```

C Zusammenfassung in deutscher Sprache

Ein Großteil der heute digital verfügbaren Informationen liegt in Form natürlichsprachlicher Texte vor. Das Ziel der *Informationsextraktion* (IE) ist es, bestimmte gewünschte Informationen aus solchen Texten zu extrahieren und in einer Form abzuspeichern, die strukturierte Abfragen ermöglicht (im Gegensatz zum *Information Retrieval*, wo die Suche nach Dokumenten und Dokumentfragmenten im Vordergrund steht).

In dieser Dissertation wird ein trainierbares statistisches Informationsextraktionssystem entwickelt. Anders als bisherige Ansätze kann unser System *inkrementell trainiert* werden, was den menschlichen Trainingsaufwand verringert.

Das System ist als generisches Framework konzipiert – alle Bestandteile des klassifikationsbasierten Informationsextraktionsmodells können unabhängig voneinander modifiziert und ausgetauscht werden. Der systematische Austausch einer der Komponenten (der Tagging-Strategien) wird im Rahmen der Arbeit untersucht.

Zur Verbesserung der Extraktionsqualitität werden verschiedene neue Informationsquellen untersucht. Die Verwendung reichhaltiger Kontextrepräsentationen auf Basis von Baumstrukturen ermöglicht es uns, neben semantischen und linguistischen Informationen auch die Dokumentstruktur als Informationsquelle zu erschließen. Um die verschiedenen und teilweise widersprüchlichen Strukturen in eine einheitliche Baumstruktur zu bringen, entwickeln wir einen Verschmelzungsalgorithmus für XML, der Verschachtelungskonflikte und andere Fehler beheben kann.

Als Kern des klassifikationsbasierten Ansatzes führen wir einen generischen Klassifikationsalgorithmus (Winnow+OSB) ein, der Online Learning mit einer neuen Art erweiterter Bigramme verbindet. Wir zeigen, dass dieser Algorithmus außer für Informationsextraktion auch für andere Anwendungen wie Textklassifikation geeignet ist – so erzielte er im Spamfilter-Wettbewerb der *Text REtrieval Conference (TREC) 2005* eines der beiden besten Ergebnisse.

Die Arbeit beinhaltet eine ausführliche Evaluation unseres Extraktionssystems, die zeigt, dass es mit anderen modernen Verfahren vergleichbare oder bessere Ergebnisse erzielt. Wir untersuchen dabei auch den Einfluss verschiedener Faktoren und Informationsquellen auf das Gesamtsystem, mit dem Ergebnisse, dass alle eine positive Rolle spielen. Weiterhin wird die Nützlichkeit des von uns vorgeschlagenen interaktiven inkrementellen Trainings gemessen; dabei bestätigt sich, dass der menschliche Trainingsaufwand auf diese Weise stark reduziert werden kann. Ergänzend zur quantitativen Evaluation analysieren wir die auftretenden Fehler und ihre mutmaßlichen Ursachen, was ein besseres Verständnis von Verbesserungsmöglichkeiten und vermutlich eher grundsätzlichen Beschränkungen der Informationsextraktion ermöglicht.

Erklärung

Ich versichere, dass ich die vorliegende Dissertation auf Grundlage der in der Arbeit angegebenen Hilfsmittel und Hilfen selbständig verfasst habe.

Berlin, den 16. Februar 2007

(Christian Siefkes)