Combining robust parsing and lexical acquisition in the XDOC system

Dietmar Rösner

Otto-von-Guericke-Universität Magdeburg
Institut für Wissens- und Sprachverarbeitung
P.O. Box 41 20, D–39016 Magdeburg, Germany
roesner@iws.cs.uni-magdeburg.de

Abstract

The project XDOC – XDOC stands for XML based tools for document processing – aims at delopping a workbench with modules that can be exploited for the processing of electronically available documents. In this paper we concentrate our discussion on XDOC's robust parsing strategies and how they are integrated with and exploited for lexical acquisition.

1 Introduction and background

The project XDOC – XDOC stands for XML based tools for document processing – aims at delopping a workbench with modules that can be exploited for the processing of electronically available documents ([5]).
We are experimenting with a number of corpora that are comprised of what we call *'realistic documents'*, i.e. texts used in everyday contexts. These include:

- emails
- technical documentation
- patient information accompanying drugs
- texts from medical textbooks (e.g. on anatomy)
- legislative texts.

In most cases we work with documents given in either English or German.
In spite of differences between these applications there are as well many commonalities. This motivates the attempt to create a set of tools for those aspects of document processing that are relevant in all or at least most applications.
There are a number of such toolsets available for English (e.g. GATE ([2]) and LT XML ([4])). Therefore our own development has concentrated on work for a set of tools for German. This is in line with the principle to try to make best use of available ressources and to avoid duplication of work.

2 Design rationale

Many of our design decisions are best motivated by our working assumptions about available and necessary ressources for successful processing. This includes as well deliberate decisions about ressources that may be incomplete.
Other decisions take the nature of our target documents into account. Realistic texts show many linguistic phenomena that are different from textbook treatments of grammar at least in the frequency distribution of certain phenomena.
Another aspect is that the value of a toolbox like XDOC is not created by a single powerful algorithm but constitutes itself rather through a multitude of

small modules with – taken in isolation – simple tasks to perform that can be easily combined and adapted to different applications.
This makes *interoperability* a major concern for the design.
We therefore decided that all components of the XDOC system – irrespective of their internal data structures – will operate on texts (files or strings) that are tagged in XML style ([1]) and that all components deliver their results in the same format. This allows to flexibly combine the components and to realise complex functionality – like with 'piping' in UNIX – through the combination of elementary components. A similar approach has proved to be successful in the LT XML project of the Language Technology Group in Edinburgh ([4]).
Another design decision is to aim for *robustness* in the sense that when one module has not all information available for a full result it should not fail but create the best intermediate result possible. Other modules might later be able to acquire the missing information using additional constraints available to them.
This is best illustrated with the treatment of unknown kexical items in the tagger and parser of XDOC:
XDOC's approach to lexical acquisition is based on the observation that lexica for *open* word classes will always be *incomplete*. This is not only due to neologisms in the general vocabulary but is primarily true for sublanguages in e.g. engineering and science. Whenever you change the domain of discourse in such fields it is likely to come across lexical items not already in the lexicon.
On the other hand there are morphological tools with a *complete coverage* of the *closed* class lexical items of German available. We decided to use the system MORPHIX ([3]) which has this property. MORPHIX comes as well with a lexicon of *open* class items and can analyse and generate their inflected forms. But this latter lexicon was filled – except for German irregular verbs, which are systematically covered – according to the needs of the various projects using MORPHIX and has neither a representative nor in any sense complete coverage. In other words: a tagger based on MORPHIX will in many cases not be able to classify unknown *open* word class items, but the parser can rely on the correctness of the classification of *closed* class items and exploit this information (cf. below).

3 System overview

The tools in the current state of XDOC can roughly be grouped into the following categories:

- preprocessing,
- language identification,
- structure detection,
- POS tagging,
- parsing.

These tools are of a more general nature and the resp. tasks show up in many document processing applications.

Preprocessing tools These are needed to map documents that are already electronically available into a normalised form that is assumed in the following processing steps. This primarily comprises 'deformatting' with standard tools like LaTeX-to-ASCII converters and the conversion from a number of different character sets into a normalised character set.

Language identification Some of our corpora are multilingual, e.g. the collection of emails from our own email traffic. The documents in this corpus are mostly written in English or German, a few are in other languages like French. In such situations automatic language identification is necessary so that for subsequent processing appropriate language dependent tools can be activated.
In XDOC the primary language of a document is determined on the basis of the relative frequency of closed class lexical items (stop words) from the candidate languages.

Detection of structural units The general task is here to identify sequences from the unstructured stream of ASCII characters as structural units of a document and to identify their function in the document. For emails this task includes e.g. the separation of the (optional) epilogue from the text body and the detection of cited emails.

Tagging of interpunction Tagging of interpunction may be seen as part of structure detection or as part of tagging. Irrespective of whatever classification is preferred it is a prerequisite for the detection and processing of sentences and other linguistic units. Some of the difficulties with the correct interpretation of interpunction are illustrated in the following example text:

```
XDOC(269): (tag-ip-line "Punkte koennen auch in
Abkuerzungen - wie z.B. Prof. Dr. -, in email-
Adressen oder in Dezimalzahlen (wie 3.14)
verwendet werden.")
"Punkte koennen auch in Abkuerzungen - wie
<ABBR>z.B.</ABBR> <ABBR>Prof.</ABBR>
<ABBR>Dr.</ABBR> -<IP>,</IP> in email-Adressen
oder in Dezimalzahlen <IP>(</IP>wie 3.14<IP>)</IP>
verwendet werden<IP>.</IP>"
```

Tagging of word classes The POS tagger of XDOC used for German is based on the morphological component MORPHIX.
It works as follows: If the input is not already tagged with respect to interpunction, this is performed first. Then for each candidate lexical item it is checked if MORPHIX analyses exist and what the respective word classes are. If no MORPHIX word classes can be determined for a candidate string heuristics are applied that are based on properties of the candidate string and that take its relative position in relation to interpunction into account. It is for example exploited that a candidate string starting with an uppercase letter but not in sentence initial position can be classified as noun.
Here is the result of applying XDOC's tagger to a short example sentence from a technical handbook:

```
XDOC(266): (tag-text "Strebe die geringstmoegliche
Zahl unterschiedlicher Kerne an!")
"<V>Strebe</V> <MULT VAL=(\"DETD\" \"RELPRO\")>die
</MULT> <XXX>geringstmoegliche</XXX> <N>Zahl</N>
<ADJ>unterschiedlicher</ADJ> <N SRC=UC1>Kerne</N>
<MULT VAL=(\"PRP\" \"VZ\")>an</MULT><IP>!</IP>"
```

Most of the tags used for standard lexical classes (e.g. `<N>`, `<V>`, ...) and for non-standard lexical classes (e.g. `<ABBR>` for abbreviations) should be selfexplaining. `<XXX>` denotes unrecognized lexical items. Here the lexical item `'geringstmoegliche'` is neither in the lexicon nor classified with a heuristic and therefore tagged as unrecognized. `<MULT ...>` is given for lexical items with multiple readings – given as list in the attribute `VAL` – when taken in isolation. Here the lexical item `'die'` may be either a definite determiner or a relative pronoun and `'an'` may be either a preposition or a separated prefix of the verb (tagged as `VZ` for 'Verbzusatz'). When a wordclass is detected based on a heuristic the name of the latter is recorded as process information within the tag as value of the attribute `SRC` (like in `<N SRC=UC1>Kerne</N>`).
How useful is a tagger result with multiple categorisations? A first strategy is to further improve the result. We have experimented with heuristics that work on a tagger result with multiple classifications and try to improve it by disambiguating multiple readings.
For the example text the following heuristics are applicable:

- WHEN the alternative is between a definite determiner and a relative pronoun AND no comma is immediately preceding, THEN choose definite determiner.

- WHEN the alternative is between preposition, verb prefix and - optionally - a subordinating conjunction AND the candidate immediately precedes a sentence closer, THEN choose verb prefix.

This yields the following improvements:

```
XDOC(267): (improve-tagger-result *)
"<V>Strebe</V> <DETD SRC=MH1>die</DETD>
<XXX>geringstmoegliche</XXX> <N>Zahl</N>
<ADJ>unterschiedlicher</ADJ> <N SRC=UC1>Kerne</N>
<VZ SRC=MH2>an</VZ><IP>!</IP>"
```

These heuristics have been defined in a 'conservative manner' in the sense that alternative readings of lexical items should only be dismissed when the circumstances are strongly supporting. Nevertheless their application sometimes has to rely on 'tacit assumptions' about the nature of the input texts. The first heuristic for example assumes the correct usage of commata.
A second strategy for treating the tagger output is to integrate disambiguation as well as classification into the robust parsing process. This approach is discussed in the following section.

Robust partial parsing In the applications of XDOC it is necessary to be able to treat linguistic input in a robust manner.
For the syntactic analysis we therefore employ a chart parser that is adapted to handle tagging results with multiple classifications as well as with unclassified lexical items.
The core **extension of the chart mechanism** is as follows: If a grammar rule expects a lexical element from an open word class but the current position in the input contains an element tagged as unknown (i.e. <XXX>), then the parse continues with the assumption that the unclassified lexical item belongs to that word class. This assumption is recorded in the feature structure for the element.
It is of course unavoidable that during the parse with any nontrivial grammar this will yield wrong local hypotheses, but rules covering larger parts of the input in many cases serve as effective filter (cf. excerpts from a trace below. Note that the two readings found here only differ with respect to the surface cases - i.e. feature CAS with value AKK or NOM – of the complex NP, therefore repetitive part elided).

```
XDOC(29): (results2xml
     (chart-parse (tag-text
       "ein Beispiel mit unbekannten Woertern")))

 "unbekannten" assumed to belong to wordclass V
 ...
 "unbekannten" assumed to belong to wordclass ADV
 "unbekannten" assumed to belong to wordclass ADJ
 ...
```

```
"<READINGS>
<READING NO=1>
  <NP TYPE=COMPLEX GEN=NTR NUM=SG CAS=AKK>
    <NP CAS=AKK NUM=SG GEN=NTR>
      <DETI>ein</DETI>
      <N>Beispiel</N>
    </NP>
    <PP>
      <PRP>mit</PRP>
      <NP CAS=_ NUM=_ GEN=_>
        <XXX AS=ADJ>unbekannten</XXX>
        <N SRC=UC1>Woertern</N>
      </NP>
    </PP>
  </NP>
</READING>
<READING NO=2>
  <NP TYPE=COMPLEX GEN=NTR NUM=SG CAS=NOM>
    <NP CAS=NOM NUM=SG GEN=NTR>
    ...
  </NP>
</READING>
</READINGS>"
```

The XDOC parser employs a context free grammar annotated with feature structures (attribute-value pairs). The treatment of these annotations is again implemented to support robust processing: At the entry level of the chart lexical items are introduced according to the tagger classifications but annotated with possible values for their morphosyntactic features (e.g. case, number and gender for lexical categories N, ADJ, DETD, ...) if these can be provided by MORPHIX (cf. above). If no morphosyntactic features are available (because the lexical item is classified as XXX – in the example sentence: 'geringstmoegliche' – or because there is no MORPHIX lexicon entry yet – in the example: 'Kerne' classified as noun –, then underspecified feature structure are created (with the special symbol '_' for unspecified values).
During parsing agreement is e.g. treated via intersecting sets of possible feature values. Resulting feature structures are propagated 'upward' (e.g. from NP constituents to the NP) and are tested for in rules for clausal and sentential structures.
The combination of these strategies effectively restricts the number of readings in many cases. In

the example below only two possible reading 'survive' that span the whole input sentence. It is worth to note the following: For the string 'geringstmoegliche' only the interpretation as ADJective is licensed by this complete result, other possible readings – e.g. as a Verb – are filtered in this context, but could be possible in others (for other strings; please note: no prior knowledge about the interpretation of this very string is assumed to be available here). For the nominal phrase 'die .. Zahl' the other possible case value NOMinative is filtered. For the string 'unterschiedlicher Kerne' we get two readings as genitive NPs but with different, incompatible feature structures. The first is restricted to nouns with the lexical gender FEM and any value for number, the second is restricted to number PL, but allows any lexical gender.

```
XDOC(239): (results2xml
 (chart-parse-sentence (tag-text "Strebe die
  geringstmoegliche Zahl unterschiedlicher
  Kerne an!")))

 "geringstmoegliche" assumed ... wordclass V
 ...
 "geringstmoegliche" assumed ... wordclass ADV
 "geringstmoegliche" assumed ... wordclass ADJ
 ...
WARNING: no features available for constituent
  (N "Kerne" :SRC UC1)
  BUT: we assume underspecified feature values
 ...

"<READINGS>
<READING NO=1>
  <S MOOD=IMP NP-ARG1=OBJ>
    <V ROOT=streb FLEX=FIN>Strebe</V>
    <NP TYPE=COMPLEX GEN=FEM NUM=SG CAS=AKK>
      <NP CAS=AKK NUM=SG GEN=FEM>
        <DETD>die</DETD>
        <XXX AS=ADJ>geringstmoegliche</XXX>
        <N>Zahl</N>
      </NP>
      <NP CAS=GEN NUM=_ GEN=FEM>
        <ADJ>unterschiedlicher</ADJ>
        <N SRC=UC1>Kerne</N>
      </NP>
    </NP>
    <VZ>an</VZ>
    <IP>!</IP>
  </S>
</READING>
<READING NO=2>
  <S MOOD=IMP NP-ARG1=OBJ>
    <V ROOT=streb FLEX=FIN>Strebe</V>
    <NP TYPE=COMPLEX GEN=FEM NUM=SG CAS=AKK>
      <NP CAS=AKK NUM=SG GEN=FEM>
        <DETD>die</DETD>
        <XXX AS=ADJ>geringstmoegliche</XXX>
        <N>Zahl</N>
      </NP>
      <NP CAS=GEN NUM=PL GEN=_>
        <ADJ>unterschiedlicher</ADJ>
        <N SRC=UC1>Kerne</N>
      </NP>
    </NP>
    <VZ>an</VZ>
    <IP>!</IP>
  </S>
</READING>
</READINGS>"
```

4 Current state and future work

The XDOC system as described above is implemented in Allegro CommonLisp and running under UNIX on SUN and SGI workstations.

The system is currently tested, evaluated ('How robust is the processing?', 'How reliable is lexical acquisition?', 'How usable is XDOC by people not involved in the development?', ...) and further extended – not only in grammar coverage, but in functionality as well – in applications. These include:

- email processing: It is attempted to extract information from the text body that allows to help in preselection and prioritizing of emails according to user preferences.

- experiments in terminology and concept extraction from medical text books and from technical texts.

System availability

Major components of XDOC are made accessible for testing and experiments under the URL:
http://berlin.cs.uni-magdeburg.de:8000/

References

[1] Tim Bray, Jean Paoli, and C.M. Sperberg-McQueen. Extensible Markup Language (XML) 1.0. *http://www.w3.org/TR/1998/REC-xml-19980210*, 1998.

[2] H. Cunningham, Y. Wilks, and R. Gaizauskas. GATE - a General Architecture for Text Engineering. In *Proceedings of COLING-96*, 1988.

[3] Wolfgang Finkler and Günter Neumann. MORPHIX. A Fast Realization of a Classification-Based Approach to Morphology. In *Trost, H. (ed.): 4. Österreichische Artificial-Intelligence-Tagung. Wiener Workshop - Wissensbasierte Sprachverarbeitung*, pages 11–19, Berlin etc., 1988. Springer.

[4] Language Technology Group. LT XML version 1.1. *http://www.ltg.ed.ac.uk/software/xml/*, 1999.

[5] D. Rösner. XDOC – XML-basierte Werkzeuge für multilinguale Korpora. In *Multilinguale Corpora – Codierung, Strukturierung, Analyse; 11. Jahrestagung der GLDV*, pages 332–341. enigma corporation, Prag, 1999.