Simplified MMAXQL: An Intuitive Query Language for Corpora with Annotations on Multiple Levels

Christoph Müller
EML Research gGmbH
Villa Bosch
Schloß-Wolfsbrunnenweg 33
69118 Heidelberg, Germany
christoph.mueller@eml-research.de

1 Introduction

Growing interest in richly annotated corpora is a driving force for the development of annotation tools that can handle multiple levels of annotation.[1] Specialized query languages are employed for the exploitation of these corpora.

In order to make full use of the potential of multi-level annotation it is crucial that individual annotation levels be treated as *self-contained modules* which are independent of other annotation levels. This should also include the storing of each level in a separate file. If this principle is not observed, annotation data management (incl. level addition, removal and replacement, but also conversion into and from other formats) is made more difficult than necessary. Moreover, *multi-level querying* will be facilitated if annotation levels are independent of each other, because users can relate markables from all levels in a fairly unrestricted way, without having to consider representational issues that are irrelevant for their current query. This facilitates exploratory data analysis of annotated corpora for all users, including non-experts.

In our multi-level annotation tool MMAX2[2] (Müller & Strube, 2003) markable levels are independent of each other. The query language MMAXQL is rather complicated and not suitable for naive users. We present an alternative query method consisting of a more intuitive query language and an implemented method to generate MMAXQL queries from the former. The new, *simplified* MMAXQL can express a wide range of queries in a simple and compact way, including queries for discourse-level phenomena like coreference.

2 Simplified MMAXQL

A query in simplified MMAXQL consists of a sequence of *query tokens* which describe elements (i.e. either words or markables) to be matched, and *relation operators* which specify which relation should hold between the elements matched by two adjacent query tokens. Relations that can be queried include sequential, hierarchical, and associative relations. The **sequential** relation between two elements can be queried by means of the operator `before` (A ends before B begins) and `meets` (A ends when B begins)[3]. The **hierarchical** relation between two elements can be queried by means of the operator `in` (A is completely included/embedded in B) and `dom` (A completely contains/dominates B). The operators `starts` and `ends` combine the sequential and hierarchical relations, with `starts` standing for left alignment (A starts

[1] This description is based on Müller (2005).

[2] The current release version of MMAX2 can be downloaded at http://mmax.eml-research.de.

[3] This is the default operator.

when B starts and ends before B ends) and `ends` standing for right alignment (A starts after B starts and ends when B ends). In addition, **associative** relations like set membership can be queried by means of the relation operator `nextpeer` (cf. below for an example).

A query for **words** consists of regular expressions in single quotes. Each expression matches one word exactly. The query[4] `'[Yy]ou know'`, e.g. returns 59 hits in the form of 2-tuples, taking about 3 seconds to search the approx. 13.000 words of the document. *You know* is interesting in spoken dialogue because it can either have its literal meaning or be a lexicalised filled pause.

A query token for a **markable** is of the form `regExp/conditions`, where the (optional) `regExp` part specifies the text of a markable, and the `conditions` part defines matching conditions with respect to markable attributes and values. In the minimal form, a condition only specifies the name of a markable level. The sample document contains, among others, a *segment* level with 1398 markables roughly equivalent to speaker turns, and a *meta* level containing 1031 markables representing e.g. pauses, emphases, or sounds like breathing or mike noise. The query `/segment` retrieves a list of 1398 1-tuples in about 2 seconds. The query `.*pars.*/segment` returns the 3 segments which contain the string *pars* in about the same time.[5]

A more common way of query is in terms of attribute-value combinations. Simplified MMAXQL contains (optional) features which make this considerably easier. First of all, if the attribute name is unique across all markable levels, the level name can be left out, since the attribute name unambiguously points to it. Thus, a query like `/type=emphasis` can query markables from the `meta` level, granted that only one attribute of name `type` exists.[6] Furthermore, if an attribute is defined as having a closed set of possible values (as is the case for the *type* attribute on the *meta* level), and if the required value is unique across all values of all other attributes on all other levels, the attribute name can be left out as well. Thus, the above query can be reduced to `/emphasis`, which is shorter and more intuitive since what the user wants is finding cases of emphasis rather than particular attribute-value combinations. On the sample document, the query returns 265 hits in about 4 seconds.

Elements from several levels can be mapped to each other by joining query tokens using relation operators. The result of such a query is a tuple with as many columns as the query contained query tokens. In the following example, the *meta* and *segment* levels and the word level are combined in a query to retrieve instances of *you know* that appear in segments spoken by female speakers[7] which also contain a pause or an emphasis.

```
'[Yy]ou know' in (/participant={f.*} dom /{pause,emphasis})
```

The following equivalent but much more verbose and complicated MMAXQL query is automatically generated from the above:

```
let $10=segment (*participant={f.*});
let $11=meta (type={pause,emphasis});
let $22=contains($10, $11);
let $20=basedata (*basedata_text={[Yy]ou});
let $21=basedata (*basedata_text={know});
let $2=during(meets($20, $21), $22);
display $2;
```

Our ICSI corpus does not yet contain coreference annotation, but that is in the process

[4]Unless noted otherwise, all examples come from document BDB001 of the ICSI Meeting Corpus, a corpus of spoken multi-party dialogue (Janin et al., 2003). The corpus was obtained from the Linguistic Data Consortium and completely converted into MMAX2 format, preserving all original information. Reported query times were on a Pentium Mobile III/800 with 512 MB RAM.

[5]The `.*` wild cards in the latter query are required since by default a query matches whole markables only.

[6]If this condition does not hold, the attribute name can be disambiguated by prepending the level name.

[7]The first letter of the `participant` value encodes the speaker's gender.

of being added. Therefore, the following example is taken from a different corpus which consists of a part of the Penn Treebank portion of the Switchboard corpus. This corpus was converted into MMAX2 format and subsequently annotated for coreference, using a markable level with name `coref` and a markable set attribute with name `member`. On this corpus, the following query can be used to retrieve pairs of anaphors (right) and their direct antecedents (left).

```
/coref nextpeer:member /coref
```

On sample document 0013_4617 from our corpus, the query returns 51 2-tuples in about 2 seconds. The `coref` level has a `npform` attribute describing the morphological form of the expression. Thus, the query can be modified as follows to return only anaphoric *pronouns* and their direct antecedents.

```
/coref nextpeer:member /coref.npform=prp
```

This reduces the number of hits on the sample document to 32.

The corpus also contains coreference annotation for non-NP antecedents, i.e. statements or propositions that are referred back to by means of pronouns (mostly by means of *that*). Non-NP antecedents are tagged with the value `utt` (for utterances) and `vp` (for verb phrases) in the `expressionstype` attribute. Thus, pairs of anaphors and their direct non-NP antecedents (either utterances OR verb phrases) can be retrieved with the following query.

```
/coref.expressionstype={utt,vp} nextpeer:member /coref
```

This query retrieves 10 2-tuples.

A single query can contain more than 2 query tuples. The following query retrieves 3-tuples of the initial markable in a coreference set and the next two mentions by just concatenating three query tuples.

```
/coref.member=initial nextpeer:member /coref nextpeer:member /coref
```

The corresponding MMAXQL query runs as follows:

```
let $10=coref (member={initial});
let $11=coref;
let $12=coref;
let $1=next_peer('member',
       next_peer('member',$10,$11),$12);
display  $1;
```

3 Future Work

The current experimental implementation does not yet include wild cards, which is particularly inconvenient for queries using the `nextpeer` operator, because without a wild card querying a chain of `n` markables requires that many literal repetitions of the query token. Thus, future work includes adding support for wild cards on the query token level. The query language also still lacks a means to express queries like '*coref* markables that are `n` *coref* markables apart.' Finally, we are also looking into ways of further optimizing query execution.

Acknowledgements

This work has been funded by the Klaus Tschira Foundation, Heidelberg, Germany.

References

Janin, Adam, Don Baron, Jane Edwards, Dan Ellis, David Gelbart, Nelson Morgan, Barbara Peskin, Thilo Pfau, Elizabeth Shriberg, Andreas Stolcke & Chuck Wooters (2003). The ICSI meeting corpus. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, Hong Kong.

Müller, Christoph (2005). A flexible stand-off data model with query language for multi-level annotation. In *Proceedings of the Interactive Posters/Demonstrations session at the 43rd Annual Meeting of the Association for Computational Linguistics*, Ann Arbor, Mi., 25-30 June 2005. To appear.

Müller, Christoph & Michael Strube (2003). Multi-level annotation in MMAX. In *Proceedings of the 4th SIGdial Workshop on Discourse and Dialogue*, Sapporo, Japan, 4-5 July 2003, pp. 198–207.