The MATE Workbench

[1]Laila Dybkjær, [1]Morten Baun Møller, [1]Niels Ole Bernsen, [2]Jean Carletta,
[2]Amy Isard, [3]Marion Klein, [2]David McKelvie, [4]Andreas Mengel
1: NIS, Odense University, Science Park 10, 5230 Odense M, Denmark.
2: HCRC, 2, Buccleuch Place, Edinburgh EH8 9LW, UK.
3. DFKI, Stuhlsatzenhausweg 3, D-66123 Saarbrücken, Germany.
4: IMS, Stuttgart University, Azenbergstraße 12, D-70174 Stuttgart Germany.

Abstract

This paper provides a brief overview of the MATE Workbench which is a set of tools in support of annotating and handling data from spoken dialogue corpora.

Introduction

A major problem faced by developers of spoken language dialogue systems and other language engineering products concerns the re-use of corpora or tools in different development or customisation projects. So far, most projects have had to either develop the necessary resources from scratch or to acquire resources from previous projects followed by a painstaking adaptation process. Both approaches are time-consuming and often inefficient.

The MATE project (http://mate.nis.sdu.dk) aims to facilitate the re-use of language resources by addressing the problems of creating, acquiring and maintaining spoken language corpora. On the basis of results from projects world-wide on spoken dialogue annotation and tools (Klein et al. 1998), MATE has developed a standard framework for the annotation of spoken dialogue corpora at multiple levels, including prosody, (morpho-)syntax, co-reference, dialogue acts, and communication problems, as well as the interaction among the levels (Dybkjær et al. 1998). MATE proposes state-of-the-art best practice coding schemes for its annotation levels and is currently completing the first version of a workbench, i.e. a set of integrated tools, in support of the annotation framework and the best practice schemes (Isard et al. 1998).

1 The MATE Workbench

The MATE Workbench makes it possible to produce and exploit corpora more efficiently, and with greater accuracy and consistency. The Workbench builds upon a widely accepted encoding framework and formal coding language (XML), broad-coverage best practice coding schemes for the coding levels addressed, coding import/export mechanisms, an easy-to-use interface for coding and for adding new coding schemes, and emphasis on coding best practice procedures.

The MATE Workbench includes the following functionalities:

- acquisition and manual or semi-automatic annotation and modification of data using the MATE annotation framework;
- presentation and visualisation of spoken dialogue corpora and annotations at different levels, according to user-defined partial views;
- extraction and retrieval from annotated corpora according to any combination of constraints from both the transcribed dialogue text and any type of annotation;
- statistical procedures for determining inter-coder consistency, frequency of object language phenomena etc.;
- import/export of annotated data and easy integration of results of external tools.

The Workbench is implemented in Java to make it platform-independent. It has a modular architecture that facilitates updates and addition of new tools and annotation schemes by other users. This will compensate for the unavoidable limitations of Workbench functionality as regards, e.g., the ability to import an unlimited range of corpus file formats or provide an unlimited number of annotation schemes. The demo will present the architecture of the Workbench and demonstrate the functionalities implemented so far. In the following we briefly describe the full set of Workbench functionalities.

2 Workbench functionalities and interface

The Workbench comes with a simple and basic coding module for orthographic transcription. A sound window is available to support the transcription process. One or more best practice coding modules are provided for each of the MATE coding levels including cross-level coding. Roughly speaking, a coding module includes or describes everything that is needed in order to perform a certain kind of markup of spoken language corpora. A coding module prescribes what constitutes a coding, including the representation of markup and the relations to other codings (Dybkjær et al. 1998). Users may specify their own coding modules either by opening existing ones, modifying them and saving them under a new name (the coding modules provided as part of the Workbench should not be changed), or by creating completely new coding modules by filling in an empty coding module form which is provided by the Workbench and which has an easy-to-use interface with no programming skills required. An understanding of basic XML concepts, such as the notation of elements and attributes will be needed. By creating new coding modules users may add new coding levels to the Workbench, e.g. the user could create a coding module addressing the level of semantics.

The Workbench comes with a number of default style sheets which define how output to the user is visually represented. For instance, a corpus may be displayed in musical score format, or phenomena of interest in the corpus may be given a certain colour or shown in boldface. If the user wants to modify a style sheet or define a new style sheet, a fairly detailed understanding of XML concepts and structure is required. However, we are investigating how a more user-friendly interface to style sheet specification can be enabled.

The Workbench enables information extraction of any kind from annotated XML corpora. Using a powerful query language (Mengel and Heid 1999), the user specifies the query. The answer is a set of XML references to existing corpora. By the use of style sheets the extracted information may be exported to file formats other than XML, for instance to HTML in order to serve as input to a browser.

The query mechanism also makes it possible to extract statistical information from corpora. For instance, the user may query the number of occurrences of the token “Thursday” in a corpus or ask for the number of (marked-up) nouns in the corpus. Computation of important reliability measures, such as kappa values, will be enabled.

The basic file format used by the Workbench is XML. Support is provided for conversion from certain other file formats. For the time being, converters to XML from XLabels and BAS Partitur are available. To enable conversion from XML to other file formats, the user will have to write and add a converter to the Workbench. This is done by simply extending the ConversionTool Java class and placing the new converter in a dedicated directory. Export to file formats other than XML will be developed. Any XML file can be converted into a supported format by using style sheets. However, by adding converters in the same way as for import more advanced functionalities can be achieved than by using style sheets.

User-friendliness is a central aspect in Workbench development. The aim is that users who do not know about, or who do not want to bother with, formal TEI conformant DTD specification and XML should still be able to use the Workbench for a variety of purposes. Online help facilities will be available to the user.

Acknowledgements

The work described is being carried out on Grant No. LE4-8370, MATE (Multilevel Annotation Tools Engineering), from the European Commission’s Telematics/Language Engineering Programme. The support is gratefully acknowledged. We would also like to thank all MATE partners. Without the joint efforts of the project consortium it would not have been possible to build the MATE Workbench.

References

All MATE deliverables will eventually become available at the MATE web site at http://mate.nis.sdu.dk.

Dybkjær, L., Bernsen, N. O., Dybkjær, H., McKelvie, D. and Mengel, A. (1998) *The MATE Markup Framework.* MATE Deliverable D1.2, November.

Isard, A., McKelvie, D., Cappelli, B., Dybkjær, L., Evert, S., Fitschen, A., Heid, U., Kipp, M., Klein, M., Mengel, A., Møller, M. B. and Reithinger, N. (1998) *Specification of Workbench Architecture.* MATE Deliverable D3.1, August.

Klein, M., Bernsen, N. O., Davies, S., Dybkjær, L., Garrido, J., Kasch, H., Mengel, A., Pirrelli, V., Poesio, M., Quazza, S. and Soria, S. (1998) *Supported Coding Schemes.* MATE Deliverable D1.1, July.

Mengel, A. and Heid, U. (1999) *Query Language for Access to Speech Corpora.* Forum Acusticum, Berlin. (ASA, EAA, DEGA) 14-19 March.