

Event-coreference across Multiple, Multi-lingual Sources in the Mumis Project

Horacio Saggion* and Jan Kuper**

Hamish Cunningham* and Thierry Declerck*** and Peter Wittenburg****

Marco Puts***** and Eduard Hoenkamp***** and Franciska de Jong** and Yorick Wilks*

*University of Sheffield, United Kingdom; **University of Twente, The Netherlands

DFKI, Germany; *MPI, The Netherlands

*****University of Nijmegen, The Netherlands

saggion@dcs.shef.ac.uk - jankuper@cs.utwente.nl

hamish@dcs.shef.ac.uk - declerck@dfki.de - peter.wittenburg@mpi.nl

yorick@dcs.shef.ac.uk - fdejong@cs.utwente.nl

puts@nici.kun.nl - hoenkamp@acm.org

Abstract

We present our work on information extraction from multiple, multi-lingual sources for the Multimedia Indexing and Searching Environment (MUMIS), a project aiming at developing technology to produce formal annotations about essential events in multimedia programme material. The novelty of our approach consists on the use of a merging or cross-document coreference algorithm that aims at combining the output delivered by the information extraction systems.

1 Overview of MUMIS

The vast amount of multimedia information available and the need to access its essential content accurately to satisfy users' demands encourages the development of techniques for automatic multimedia indexing and searching. It is well known that there are no effective methods for automatic indexing and retrieving of image and video fragments on the basis of analysis of their visual features. Many research projects therefore have explored the use of collateral textual descriptions of the multimedia information for automatic tasks such as indexing, classifying, or understanding.

The Multimedia Indexing and Searching Environment (MUMIS) Project carries out index-

ing by applying information extraction to multimedia and multi-lingual information sources in Dutch, English, and German, merging information from many sources to improve indexing quality, and combining database queries with direct access to multimedia fragments on the multimedia programme. In MUMIS various software components operate off-line to generate formal annotations from multi-source linguistic data in Dutch, English, and German to produce a composite index of the events on the multimedia programme. The domain chosen for tuning the software components and for testing is football where 31 types of event (kick-off, substitution, goal, foul, red card, yellow card, etc.) need to be identified in the sources in order to produce a semantic index. The elements to be extracted that are associated with these events are: players, teams, times, scores, and locations on the pitch.

Three different off-line Information Extraction components, one per language, were developed. They are being used to extract the key events and participants from football reports and to produce XML output. A merging component or cross-document coreference mechanism has been developed to merge the information produced by the three IE systems. Keyframes extraction from MPEG streams around a set of pre-defined time marks - result of the information extraction component - is being carried out to populate the database.

Formal text
England 1 - 0 Germany Shearer (52) Bookings Beckham (42)...
Ticker
41 mins: Beckham is shown a yellow card for retaliating on Ulf Kirsten seconds after he is denied a free-kick. 40' Hoekschoop Engeland met David Beckham. Slecht getrap. Meteen maakt Beckham daarna een fout en krijgt een gele kaart.
Match
David Beckham - a muted force in attack - was shown a yellow card for a late challenge on Kirsten...
Transcription
...it's gonna be a card here for David Beckham it is yellow mmm well again his was the name in the post match headlines...
David Beckham hielt die Sohle noch drüber schauen Sie mit dem Hinterteil auch harter Einsatz gegen Kirsten und Collina zeigt ihm Gelb eine der Unarten leider von David Beckham Beckham met*x Kirsten dat is nou weer dom wat die Beckham doet ja zal ie dat dan nooit leren Kirsten overdrijft nu hoor maar Kirsten gaat 't duel in geeft een zet en dan reageert Beckham op deze manier in ieder geval krijgt ie dan weer geel

Table 1: Different accounts of the same event in different languages

The on-line part of MUMIS consists of a state-of-the-art user interface allowing the user to query the multimedia database (e.g., “The fouls committed by Beckham”). The user is first presented with selected video keyframes as thumbnails that can be played obtaining the corresponding video and audio fragments. Here, we will show the information extraction components, the merging algorithm, and the user interface.

2 Information Extraction from Multiple, Multi-lingual Sources

Information extraction is the process of mapping natural language into template-like structures representing the key (semantic) information from the text. These structures can be used to populate a database, used for summarization purposes, or as a semantic index like in the MUMIS project. Multi-lingual IE has been tried in the M-LaSIE system (Gaizauskas et al., 1997), we differ from it in that MUMIS has three different IE systems, yet they all share the domain ontology as in M-LaSIE.

Sources of information in MUMIS are: formal texts, tickers, comments, and audio transcriptions (see Table 1). Formal texts, like *html* tables with lists of players, or statistics on a particular match provide accurate information on the more relevant events (i.e., result, goals), but hardly ever contain enough information for indexing the whole match.

Tickers provide a detailed account on each of the events, but the temporal information provided by them is far from been exact (minute 40 can be either 39, 40, or 41). Matches lack detailed temporal information and comments combine information from the actual match with references to related matches (i.e., how a particular player performed in the previous match). Automatic transcriptions contain many errors, yet they provide exact temporal information attached to each token.

IE from English sources is based on the combination of GATE components for finite state transduction (Cunningham et al., 2002) and Prolog components for parsing and discourse interpretation (Saggion et al., In press). The analysis of formal texts and transcriptions is being done with finite state components because the very nature of these linguistic descriptions make it appropriate the use of shallow NLP techniques. For example, in order to recognise a *substitution* in a formal text it is enough to perform identification of players and their affiliations, time stamps, perform shallow coreference and identification of a number of regular expression to extract the relevant information. Complex linguistic descriptions (i.e., tickers) are fully analysed because of the need to identify logical subjects and objects (e.g., “he is replaced by Ince”) as well as to solve pronouns and definite expressions (e.g., “the Barcelona striker”) relying on domain knowledge encoded in the ontology of the domain. Information extraction rules operate on logical forms produced by the parser and use the ontology to check constrains. In an evaluation of the IE task on formal texts, we have obtained combined f-measure between 70% and 90% for the subset of events goal, substitution, yellow card, red card, own goal, and penalty.

The Dutch IE system performs tokenisation, lexical lookup, and HMM-POS disambiguation, and morphological analysis using the Xerox Xelda toolkit. These tools produce tokens annotated with lexical and morphological information. A domain-specific lexical lookup is performed in order to identify domain verbs and names of players and their attributes. Shallow parsing is applied to the annotated tokens by using a set of context-free grammar rules that have been specified to identify the relevant events. A stack of player names is also

used to help identify missing referents.

IE from German sources consists of the following four major components: shallow linguistic components (tokenisation, morphological analysis, chunking and shallow parsing including named entity recognition and identification of grammatical functions and reference resolution); domain-specific template definition component implementing the MUMIS ontology; domain lexicon which is used to relate natural language expressions with template definitions; and template generation and filling component that uses the domain lexicon and linguistic output of the first step as a guidance to fill-in the templates. The system takes advantage of the information extracted from formal texts (e.g., lists of players) in order to carry out the analysis of tickers.

3 Merging or Cross-document Event Coreference

The merging component in MUMIS combines the partial information as extracted from various sources, such that more complete annotations can be obtained. Information extraction and merging from multiple sources has been tried in the past (Radev and McKeown, 1998) but only for single events, the novelty of our approach consists on applying merging to multiple-events extracted from multiple sources.

As an example consider the following situation (Netherlands-Yugoslavia match): One of the IE components extracted from document A that in the 30th minute of the match a free-kick was taken, but did not discover who took it. It did find the names of two players, though: Mihajlovic (a Yugoslavian player) and Van der Sar (the Dutch keeper). From document B a save in the 31st minute was extracted by the IE component, and the names of the same two players were recognised. From these two results it now can be concluded that it was Mihajlovic who took the free-kick, and that Van der Sar made the save, thus giving a more complete picture of what happened in the 30-31st minute of the match.

It is a first task of the merging component of the MUMIS project to find out which events from the various documents should be combined such that more complete information can be derived. The

person who produced the natural language text from which events are extracted, acts as a “semantic filter”: the events he/she described are understood to belong together in the same *scene* (groups of events semantically related) if they are mentioned in the same textual fragment. For example, if the same players are mentioned in two different but close (in time) textual fragments, then the events accounted for in those fragments could be connected under specific constraints.

The merging program consists of the following steps:

1) Bi-document alignment: given two source documents A and B, every scene from A is checked for compatibility with every scene from B. In determining the strength of a possible connection between two scenes, various aspects play a role: number of common player names, distance in time, etc. First, the program calculates the strength of all bindings between all pairs of scenes from documents A and B respectively. Suppose that the binding strength between a scene SA from document A and a scene SB from document B is the strongest, then the program concludes that these two scenes are about the same episode in the match, and the combination is confirmed. Choosing the combination rules out certain other combinations from the two documents A and B, e.g. combinations between scenes from document A which are before scene SA and scenes from document B which are after scene SB are eliminated. This process is repeated recursively until all possible combinations between scenes from documents A and B are fixed.

2) Multi-document alignment: the above process is performed on every pair of documents, thus yielding pairs of scenes. The next step is to build sets of scenes which are connected as follows. Create a set consisting of any scene, and add all scenes to this set which are connected to this scene by the process from step 1. Repeat this for all these newly added scenes recursively until no new scenes are found which should be added to the set. This set naturally forms a (connected) graph of combined scenes. Notice that the graph need not be complete, i.e. not every pair of scenes in the graph needs to be connected. In fact, scenes may be incompatible and neverthe-

less occur in the same graph through a sequence of intermediate scenes. Since a graph is supposed to contain scenes from various documents which all are about the same episode during the match, a graph should not contain such scenes which are incompatible in that sense. In order to exclude such scenes from a graph, the program splits a graph into complete subgraphs, such that only graphs remain in which all scenes are connected to all other scenes. This splitting up again is based on the strongest connections in a given graph.

3) Unification: the partial events from the various scenes in a given graph are combined and empty slots are filled in. At this point several (semantical) rules expressing domain knowledge are used. There are several kinds of rules to be used at this point. First, event internal rules describe which events are possible (i.e., a keeper will not take a corner). Second, event external rules express possible combinations of events (i.e., a player shooting at goal will belong to the other team as a player who blocks this shot). As a result, more completely filled in events are produced.

4) Ordering: finally the events inside such a scene have to be put into the correct order. For example, a shot on goal in the same scene as a goal typically will take place before that goal and not after. For this ordering process scenarios are used.

The output produced by the merging algorithm, which contains temporal information, is used on the one hand to extract JPEG keyframes images that serve for quick inspection in the user interface, and on the other hand to index the multimedia database. The software used for off-line keyframe extraction from MPEG1 movies is Spikes: it takes a movie file, a list of times stamps, and the size of the keyframe and produces a list of keyframes.

4 User Interface

The user interface allows the user to enter formal queries to the MUMIS system. The interface makes use of the lexica in the three target languages and domain ontology to assist the user while entering his/her query. The hits of the query are indicated to the user as thumbnails in the storyboard together with extra information about each

of the retrieved events. The user can select a particular fragment and play it.

5 Conclusion

The huge amount of multimedia information accessible directly to the end users require a new generation of tools to provide "intelligent" access to specific information. MUMIS is the first multimedia indexing project which carries out indexing by applying information extraction to multimedia and multi-lingual information sources, merging information from many sources to improve the quality of the annotation database, and combining database queries with direct access to multimedia fragments.

Acknowledgements

MUMIS is an on-going EU-funded project within the Information Society Program (IST) of the European Union, section Human Language Technology (HLT). Project participants are: University of Twente/CTIT, University of Sheffield, University of Nijmegen, Deutsches Forschungszentrum für Künstliche Intelligenz, Max-Planck-Institut für Psycholinguistik, ESTEAM AB, and VDA.

References

- H. Cunningham, D. Maynard, K. Bontcheva, and V. Tablan. 2002. GATE: A framework and graphical development environment for robust NLP tools and applications. In *ACL2002*.
- R. Gaizauskas, K. Humphreys, S. Azzam, and Y. Wilks. 1997. Concepticons vs. lexicons: An architecture for multilingual information extraction. In M.T. Paziienza, editor, *SCIE-97*, LNCS/LNAI, pages 28–43. Springer-Verlag.
- R. Radev and K.R. McKeown. 1998. Generating Natural Language Summaries from Multiple On-Line Sources. *Computational Linguistics*, 24(3):469–500, Sept.
- H. Saggion, H. Cunningham, K. Bontcheva, D. Maynard, O. Hamza, and Y. Wilks. In press. Multimedia Indexing through Multi-source and Multi-language Information Extraction: The MUMIS Project. *Data & Knowledge Engineering Journal*.