# Time Extraction from Real-time Generated Football Reports

**Markus Borg**
Department of Computer Science
Lund University, LTH
d02mbr@student.lth.se

## Abstract

This paper describes a system to extract events and time information from football match reports generated through minute-by-minute reporting. We describe a method that uses regular expressions to find the events and divides them into different types to determine in which order they occurred. In addition, our system detects time expressions and we present a way to structure the collected data using XML.

## 1 Introduction

Real-time football reports are an increasingly popular way to describe what happens during a football game. A reporter working on covering the match continually writes usually one or two sentences at a time. Whenever an interesting event happens (a goal scoring opportunity, an injury, a booking etc.), a brief description is presented and often the time is given. As the sentences are produced, people interested in what happens can for instance conveniently follow this on the Internet. Examples of such services include *Aftonbladet*[1], a Swedish newspaper and the *UEFA*[2] homepage, an example in English with so called minute-by-minute reporting.

There are also other ways to present the report. People can subscribe to a game and get text messages directly to the mobile phone. The Swedish newspaper *Helsingborgs Dagblad*[3] provides this, but there are numerous other examples. Traditionally many viewers have also followed the latest results through teletext using a normal TV set in a similar way.

The objective of this work is to discover the various events within the texts, analyze them, and order them. When submitted in addition to the text, this information could then be presented to the viewers in the form of chains of events. It would then be possible to use this data on an arbitrary platform according to user preferences. Some users may be interested in viewing a short graphical version of the action on the display of the mobile phone, while others might want to collect statistical data on a PC.

Section 2 discusses related work. In section 3, the corpus used in this project is described. In section 4, we describe how the output data is structured. Section 5 presents how the time expressions and events are found, section 6 shows how the links between those are determined. In section 7, we present the results of our evaluation. Finally, section 8 draws some conclusions and outlines directions for future work.

## 2 Previous work

There has been much research conducted on the representation of time and events and their temporal relations. Relevant recent papers include Lapata and Lascarides (2004), which like this paper focuses on ordering of events within sentences. They propose a data intensive approach to automatically capture im-

[1] The largest daily newspaper in Scandinavia with a sport section offering reporting in real-time. www.aftonbladet.se

[2] The administrative and controlling body for European football with a homepage offering minute-by-minute reporting. www.uefa.com

[3] A local newspaper in Swedish published in Helsingborg. Offers live football coverage via SMS. www.hd.se

**Joakim Nivre, Heiki-Jaan Kaalep, Kadri Muischnek and Mare Koit (Eds.)**
**NODALIDA 2007 Conference Proceedings, pp. 37–43**

plicit temporal information, relying on a probabilistic model. Machine learning techniques have been used by different groups to determine temporal relations in natural language texts. Mani et al. (2006) achieved comparably favourable results using decision trees.

There has also been research on extraction of time information in Swedish, Berglund et al. (2006) presents a way to detect time expressions and events from authentic newspaper articles in the traffic accident domain. It is part of the Carsim system (Johansson et al., 2005), which converts textual descriptions of accidents into animated three-dimensional scenes. Another project about information extraction in the domain of football is SOBA (Buitelaar et al., 2006). SOBA automatically extracts information from different sources on web pages, such as tables, texts, and image captions.

Our work presents a way to extract time information from one source of football reports generated through minute-by-minute reporting, which is the main novelty of this paper. The central claims are that regular expressions, although simple, still can be adequate for the task of extracting temporal information from limited closed-domain texts and that dividing events into types is instrumental in guessing in which order they occurred. A preliminary system was implemented to evaluate those claims.

## 3 Corpus description

The texts we worked with come from an online football management game called *Hattrick*[4]. It is currently the biggest game of its kind with close to one million active players in January 2007. Every player takes on the role as manager for a team and plays one or two games each week, which results in a huge amount of available reports in the database. At this time, the reports are available in 40 languages.

We have chosen these texts because of the availability and the fact that the variety of expressions fits very well to test the system. The texts are not too simple however, since sentences are generated as results from 170 various events. Each event has on average five different wordings, resulting in a vocabulary suitable for this project (Henriksson, 2007).

[4] An online, browser-based, football management game developed and based in Sweden. Has been running since August 30, 1997. www.hattrick.org

Table 1: Example of a match report

Efter 18 minuters spel bröt jublet lös då Nicolas Jullien kom igenom gästernas mittförsvar och dundrade in 1 - 0 för Rydebäcks. Daniel Fridquist i Rydebäcks tilldelades efter 20 minuter gult kort för osportsligt uppträdande. I den 22:e matchminuten fick gästernas mittförsvar se sig rundat av Mikael Martinsson som slog in 2 - 0 för Rydebäcks. I den 26:e minuten fick Östen Sörensson i Nynäshamns gult kort när han gick med dobbarna före in i en duell. Rydebäcks tvingades samtidigt till ett byte eftersom John Hörnsten inte kunde fortsätta efter den omilda behandlingen. Alex Lunenburg fick kliva in i hans position. 2 - 0 var ställningen i halvtid. Halvleken dominerades av Rydebäcks som övertygade med ett 55-procentigt bollinnehav.

*English version of the same report:*

In the 18th minute cheers broke out as Nicolas Jullien found his way through the guests' central defence, clipping the 1 - 0 goal in for Rydebäcks. Daniel Fridquist of Rydebäcks received a yellow card in the 20th minute for unsportsmanlike behaviour. In the 22nd minute of the match, the visitors' central line of defence had to look on as Mikael Martinsson dashed through, knocking home 2 - 0 for Rydebäcks. In the 26th minute, Nynäshamns's sten Sörensson received a yellow card for going into a challenge studs first. Rydebäcks were forced to a substitution as John Hörnsten couldn't continue playing due to the rough treatment, forcing Alex Lunenburg to come in from the sidelines. 2 - 0 was the halftime score. The forty-five minutes were dominated by Rydebäcks, with an impressive 55 percent possession of the ball.

Hattrick offers vivid texts. However, the limited variation in style made the hand crafting of football-related regular expressions tractable. An extract of a match report is shown in Table 1.

## 4 Annotation scheme

To get a useful information exchange, it is important to structure the data in a good way. For this task we have used a subset of TimeML (Pustejovsky et al., 2003) with some modifications. It is a robust specification language for events and temporal expressions in natural language. The full complexity of TimeML was not suitable at this stage of our project, therefore we have decided to work with the most useful parts

and add a football-related attribute. Our system annotates absolute time expressions, events and time links to represent the necessary information.

The absolute time expressions are represented by TIMEX3 elements. Each element contains two attributes: tid (unique ID number) and type (so far always TIME). As in this example:

```
<TIMEX3 tid="t3" type="TIME">
den 19:e matchminuten</TIMEX3>
```

The various events are annotated as EVENT elements. Each element has three attributes: eid (unique ID number), class (OCCURRENCE or STATE), and type (IDLEBALL, PREFINISH, FINISH, SAVE, or OTHER). The elements of class STATE have type OTHER. The elements of class OCCURRENCE have one of the other types, describing the event that took place on the field. As in this example:

```
<EVENT eid="e4" class="OCCURRENCE"
type="FINISH">skalla in</EVENT>
```

The links between time expressions and events are represented by TLINK elements (time links). Links between a time expression and an event have the attributes: time (tid of the TIMEX3), event (eid of the EVENT), and type (which is set to DURING in all cases, all events during the same minute of the game get this). Links between two events have the attributes: sevent (eid of source event), tevent (tid of target event), and type (so far always BEFORE). This means that the source event happens before the target event. Example:

```
<TLINK sevent="e5" tevent="e6"
type="BEFORE"/>
```

The root node is <TimeML> and the first child is <Text>. This element has one <s> child for each sentence in the report. Every <s> element contains text nodes and possibly <TIMEX3> and <EVENT> elements. The <TLINK> elements, if present, follow after the <Text> element. Table 2 shows an example of a short match report annotated with our scheme.

Table 2: Example of XML output

```
<?xml version="1.0"?>
<TimeML>
 <Text>
  <s>Efter<TIMEX3 tid="t1" type="TIME">7
  minuters spel</TIMEX3> blev publiken som
  galen efter att Mats Aronsson <EVENT
  eid="e1" class="OCCURRENCE"
  type="PREFINISH"> kom igenom bortalagets
  backlinje</EVENT> och <EVENT eid="e2"
  class="OCCURRENCE" type="FINISH">dundrade
  in</EVENT> 1 - 0 for Fortuna. </s>
  <s>Daniel Malmsten i Fortuna tilldelades
  <TIMEX3 tid="t1" type="TIME">efter 12
  minuter</TIMEX3> gult kort efter farligt
  spel. </s>
  <s>I <TIMEX3 tid="t2" type="TIME">den 25
  :e matchminuten</TIMEX3> fick bortalagets
  mitt<EVENT eid="e4" class="OCCURRENCE"
  type="PREFINISH">forsvar se sig rundat
  </EVENT> av Jonas Storm som <EVENT
  eid="e3" class="OCCURRENCE" type="FINISH">
  slog in </EVENT> 2 - 0 for Fortuna. </s>
  <s>I <TIMEX3 tid="t3" type="TIME">den 29:e
  minuten</TIMEX3> fick John Evans i Klippan
  gult kort efter en vansinnig tackling.
  </s>
  <s>Fortuna tvingades samtidigt till ett
  byte eftersom Stefan Blomdahl inte kunde
  spela vidare efter den omilda
  behandlingen. </s>
  <s>Dieter Fieback fick kliva in i hans
  position. </s>
  <s><EVENT eid="e5" class="STATE"
  type="OTHER"> Det stod 2 - 0 i pausvilan
  </EVENT>. </s>
  <s>Halvleken dominerades av Fortuna
  som overtygade med ett
  55-procentigt bollinnehav. </s>
 </Text>

 <TLINK time="t1" event="e1" type="DURING"/>
 <TLINK time="t1" event="e2" type="DURING"/>
 <TLINK time="t2" event="e4" type="DURING"/>
 <TLINK time="t2" event="e3" type="DURING"/>
 <TLINK sevent="e1" tevent="e2"
 type="BEFORE"/>
 <TLINK sevent="e4" tevent="e3"
 type="BEFORE"/>

</TimeML>
```

## 5 Detection of absolute time expressions and events

The test application has been implemented in Java and heavily uses the built-in package for regular ex-

pressions. In total, the regular expressions take less than 50 lines of code and there are probably many possible optimizations left to be done. The usage of regular expressions makes the program very fast, the analysis of the text can be made without noticeable delay.

### 5.1 Finding time expressions

The program finds absolute time expressions, for instance *"in the 16th minute"* in order to put those on a timeline. This linear dimensional conception of time is not necessarily the best choice for representing the time events (Moens and Steedman, 1987), but for the 90 minutes of a football game, we considered it suitable. Relative time expressions, for example *"5 minutes later"* are not considered at all. In our corpus, we observe only a limited number of ways to express absolute time. Two lines of code were required to get a very good recall. An example of a regular expression include:

```
(I|i) (den)? [0-9]+:e
(match|spel)?minuten
```

### 5.2 Finding events

Events on the football field are described in numerous ways to make the text interesting to the reader. Every reporter has one personal style of writing and since the texts in Hattrick have developed during many years and different people have been involved, the finding of the events proved to be more of a challenge. The diversity of the football language used demanded about 45 lines of regular expressions. By grouping those according to the different types described in section 4, the event is given its type at the same time as it is detected in the text. Three examples of regular expressions are shown below.

```
(reducera|kvittera) till
[0-9]+ - [0-9]+
(komma|tagit sig) igenom
drygade [\\w]+ ut (sin ledning|
ledningen till [0-9]+ - [0-9]+)
```

## 6 Time links

The detection of time links between events or between events and time expressions is of course critical to this application. If not an unquestionable majority of the time links are correct, the resulting output file cannot be considered useful. Links between time expressions and events are always inserted in the same way, but we have tried two different approaches between events.

### 6.1 Connecting time expressions and events

Since all absolute time expressions in the football reports we have observed have expressed a certain minute of the game and since the reports are generated in real-time, the following strategy is used. If the sentence contains a time expression, all events within the same sentence are considered to occur during this time. We have not encountered any examples contradicting this so far. Therefore, one `TLINK` of the type `DURING` is added for every event in the sentence. Cases of multiple absolute time expressions within the same sentence have not been encountered and are not treated in special ways.

### 6.2 Ordering events

The fact that the text is generated in real-time, means that the later the sentence was written, the later the contained events happened. Consequently, the task is to find the chain of events within the actual sentence being processed. The chronological order of a football report written after the final whistle is much harder to determine since this property seldom is the case.

In this project, it was assumed that the events involved with goal scoring opportunities always could be ordered in a linear fashion. If it is said that a striker scores a goal and the team got the equalizer, then the goal is considered to happen before the result changes. Other approaches could be used according to taste, but here those events are not thought as simultaneous and the time links are inserted accordingly. The first event is given a time link to the next one and so on until the last event has been reached.

The basic assumption used to implement the ordering was that the different events within a sentence appear in the text in the same order as they happened during the game. This very simple approach is used as the baseline in the evaluation in section 7.

The second strategy implemented, instead utilizes the divison into the six different types, described in section 5.2. The types are described in Table 1.

Table 3: Types of events

| # | *Type* | *Example* |
|---|---|---|
| 0 | RESULT-CHANGE | Team taking the lead, scored another goal etc. |
| 1 | SAVE | Keeper saves, defender blocks etc. |
| 2 | FINISH | Shots, touches etc. towards the goal |
| 3 | PREFINISH | Passes, crosses, rushes etc. |
| 4 | IDLEBALL | Set pieces, keeper throwing the ball etc. |
| 5 | OTHER | All events of the class "STATE" |

The types are then considered to always follow a certain order regarding each other. The types given high numbers happen before the lower ones. If multiple events of the same type are present, they get time links in the same order as they appear in the text.

## 7 Evaluation

This section contains the results of an evaluation of the system, aimed at testing the recall and precision of the regular expressions used. The two different strategies of inserting time links were also tested. Since the size of the experiment is small, the results can only be taken as indicative.

### 7.1 Experimental setup

To make our application able to handle enough football-related expressions, we used 25 different texts in the training set, from which we crafted the regular expressions. The following composition of reports was used: eight reports from league games in higher divisions, seven reports from leagues in the middle, five reports from lower divisions and five reports from matches between national teams. This should ensure that reports from teams of various levels are covered by the system.

Our test set contained three reports from different teams. We selected reports from matches with 4 goals, to be certain that enough goal scoring opportunities were described in the text. Then we annotated the texts by hand in what we consider to be the correct way, by finding all expressions and detecting the correct order of events. In the end, we compared our results with the output from the system.

### 7.2 Results

We started by looking at the absolute time expressions. This proved to be an easy task and the system found all of them. We suspected this early while working with the training set. Absolute time appears to be expressed in limited ways in football reports.

As the next step, we measured how many of the events were found. The reports in total contained 53 and our system reached a recall of 79.4% with a precision of 87.5%. The recall level could be increased simply by adding more texts to the training set. The precision found however, was lower than expected and further analysis showed that some mistakes were repeatedly made. Some key words the system is looking for are used in various situations. A good example of this is the word hörna 'corner', which in Swedish is used both for the actual corner kick and when defenders or keepers save the ball by redirecting it and it passes the short line, resulting in a corner kick for the attacking team. Without a word sense disambiguation step, getting a perfect precision would be impossible. One way could be to first test if it appears after an event of the type `FINISH` or not.

Apparently, without considering parts of speech or other language characteristics, it was possible to quickly get an acceptable recall with a system entirely based on regular expressions.

The final part of the evaluation was about testing if our ideas of dividing events into certain types gave a better ordering. The three reports contained 18 sentences with multiple events, in total 47 events, suggesting that they seldom are alone in a sentence in a football report. They were divided the following way: 12 sentences had two events, one sentence had three events and five sentences had four events. Five of the events were wrongly detected, since the system treated some of the single events as two.

If the additionally found events were disregarded altogether, the baseline produced correct time links for 12 of the 18 sentences (66.7%). The strategy with the types gives a correct output between all the remaining events.

The additional events do not necessarily have to

be disregarded however, since they can be assumed to happen after the core event they were derived from. With this assumption, the result is as follows: the baseline still produced the correct result for 12 sentences (66.7%). The more complex strategy produced correct time links for 15 sentences (83.3%).

The result of the baseline shows that the events in a football report cannot be considered to happen in the same way as they appear within a sentence. We can also conclude that dividing events into those different types and assuming that passes happen before shots etc., gives a better result. The failed time links are produced in this evaluation because of failed event detection. Since some additional set pieces were introduced, they were treated as the starts of the event chains. Examples of this were shots from the penalty area (treated like penalty kicks) and the issue of corners as previously described. Still, the more complex strategy gave a significant increase in producing correct time links.

## 8 Conclusions

This paper described a way to extract time information from football reports, generated in real-time by the game engine in Hattrick. The evaluation of the system showed that if a sentence contains events, there are usually more than one. Those events cannot be expected to have happened in the same order as they appeared within a sentence written in Swedish. Although the limited set of data prevents any definitive conclusions, the work indicates that regular expressions together with type divided events can produce output well describing events on a football field. The methods should be possible to apply also on other domains with a somewhat limited vocabulary.

There are some limitations of the project. Firstly, we only consider events that have to do with goal scoring opportunities. Secondly, since the nature of real-time generated reports means that the events in the current sentence happen after the previously reported events in prior sentences, we only construct partial orderings. In this case, it means we only look at chains of events within sentences. Thirdly, no information about participants is extracted.

Further extensions could be to include also other types of events like injuries and substitutions, but we think that scoring events are more interesting to focus on at this stage. We also think it would make sense to add information about whether something actually happened or not, since this version of the system does not differentiate between “had a chance to shoot but did not” and “came through and shot”. Both shots would now be treated as the same `FINISH` type.

The next step to make the system more robust could be to include a part of speech tagger. Handcrafting regular expressions is obviously possible for limited domains, but since natural language is neither regular nor context-free, the method is not scalable for future more complex texts. However, the system is probably already good enough to be tested for simple visualization purposes of Hattrick reports.

## Acknowledgements

We would like to thank Richard Johansson and Pierre Nugues who supervised us during this project and Christian Henriksson, Language Administrator at Hattrick, for explaining internals of the report system and providing us with figures.

## References

Anders Berglund, Richard Johansson, and Pierre Nugues. 2006. A machine learning approach to extract temporal information from texts in Swedish and generate animated 3D scenes. In *Proceedings of EACL-2006*, Trento, Italy, April 15-16.

Paul Buitelaar, Phipipp Cimiano, Stefania Racioppa, and Melanie Siegel. 2006. Ontology-based information extraction with SOBA. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*, pages 2321–2324. ELRA, May.

Christian Henriksson. 2007. Personal correspondence by email. Language Administrator, Hattrick Limited (division of ExtraLives AB), Jan.

Richard Johansson, Anders Berglund, Magnus Danielsson, and Pierre Nugues. 2005. Automatic text-to-scene conversion in the traffic accident domain. In *Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence*, pages 1073–1078. Edinburgh, Scotland.

Mirella Lapata and Alex Lascarides. 2004. Inferring sentence-internal temporal relations. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*.

Inderjeet Mani, Marc Verhagen, Ben Wellner, Chong Min Lee, and James Pustejovsky. 2006. Machine learning of temporal relations. In *ACL '06: Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the ACL*, pages 753–760, Morristown, NJ, USA. Association for Computational Linguistics.

Marc Moens and Mark Steedman. 1987. Temporal ontology in natural language. In *Proceedings of the 25th annual meeting on Association for Linguistics*, pages 1–7. Stanford, California, July.

James Pustejovsky, JosCasta, Robert Ingria, Roser Saurand Robert Gaizauskas, Andrea Setzer, and Graham Katz. 2003. Timeml: Robust specification of event and temporal expressions in text. In *Fifth International Workshop on Computational Semantics*, pages 1073–1078.