Searching for Patterns in Political Event Sequences: Experiments with the KEDS Database

Klaus Kovar[1], Johannes Fürnkranz[1], Johann Petrak[1],
Bernhard Pfahringer[1], Robert Trappl[1,2], Gerhard Widmer[1,2]

[1] Austrian Research Institute for Artificial Intelligence
Schottengasse 3, A-1010 Vienna

[2] Department of Medical Cybernetics and Artificial Intelligence, University of Vienna
Freyung 6/2, A-1010 Vienna, Austria

Abstract

The paper presents an empirical study on the possibility of discovering interesting event sequences and sequential rules in a large database of international political events. We have implemented and extended a data mining algorithm, first presented by Mannila & Toivonen (1996), which is able to search for *generalized episodes* in such event databases. Experiments conducted with this algorithm on the KEDS database, an event data set covering interactions between countries in the Persian Gulf region, are described. We report some qualitative and quantitative results, and also discuss our experiences with strategies for reducing the problem complexity and focussing the search on interesting subsets of events.

To appear in *Cybernetics and Systems* 31(6), 2000.

1 Motivation

Recent years have witnessed increasing interest in the potential of data mining methods not only in business and technical applications, but also in the social sciences. As more and more empirical data are collected, the development of computer methods for searching for interesting and possibly predictive patterns in large databases becomes important.

In the political sciences, in particular, a number of databases have been established that document various aspects of international politics and its history, e.g., the CONFMAN [Bercovitch & Langley, 1993], SHERFACS [Sherman, 1988], and KOSIMO [Pfetsch & Billing, 1994] databases, to name but a few. In previous investigations, we have shown that data mining methods can indeed find predictive patterns (mostly classification rules) in such databases (see, e.g., [Fürnkranz et al., 1997, Trappl et al., 1996]).

The present study continues this line of research. Its purpose was to investigate whether interesting *temporal patterns and dependencies* can be discovered in large termporally ordered datasets. Knowledge of such temporal dependencies or action patterns might be useful, e.g., in early warning settings. We selected an international event database with rather simple structure, but with a very large number of entries (more than 300.000). From a technical point of view, we were particularly interested in the complexity aspects introduced by the sheer size of such a database, and in possible ways of reducing that complexity. To that end, we re-implemented a well-known algorithm that searches for temporal dependencies, and extended it with some facilities that allow the user to explicitly define the search space and thus influence the problem complexity. In section 4, we will present some illustrative experimental results and discuss various strategies for making such an approach computationally feasible.

2 The KEDS Database

The KEDS database [Gerner & Schrodt, 1996] is a WEIS-coded event data set that describes bilateral political events in the Arabian/Persian Gulf region (that includes countries like Iran, Iraq, Kuwait, Oman, Saudi Arabia, Yemen, and the smaller Gulf states, but also Israel, its arabic neighbors, and various major international organizations). The coded events cover political and military interactions between these parties for the period April 1979 to March 1999. The data set was generated automatically from Reuters news reports downloaded from the NEXIS data service. The data was coded with the **K**ansas **E**vent **D**ata **S**ystem (KEDS) machine-coding program into the very simple and abstract general form indicated in Figure 1.

date	*source code*	*target code*	*event code*
990101	IRN	IRQ	032
990101	IRQ	IRN	023
990102	USA	JOR	081
…………. ….	…….	…….	…….

Figure 1: Structure of entries in the KEDS database

IRN or IRQ are abbreviations for countries involved in interactions (Iran and Iraq, in this case). The first country (the field named *source code* in Fig.1) is the 'active' country, the actor who caused the event. The *second (target code*) is the 'passive' one. The event codes, which describe the actions between the two countries (or international organizations), come from a set of 110 predefined actions used in this database. For details on KEDS see [Schrodt,Gerner,1998]. The data set can be downloaded from the KEDS web site http://www.cc.ukans.edu/~keds/**.**

In our experiment, we worked with two versions of the database. The first version consists only of events coded from the lead sentences of the Reuters reports and has 57131 entries. The second version contains events derived from the full news agency articles and comprises 304402 entries. Both databases cover the time from 15 April 1979 to 31 March 1999.

3 Searching for Temporal Dependencies in the KEDS Database

As mentioned in the introduction, the goal of this research was to investigate whether interesting temporal event patterns, possibly useful in an early warning setting, could be extracted from this database with the help of data mining methods. The patterns will take the form of temporal dependency rules (or episode rules – see section 3.1.2 below). The algorithm we have implemented for this purpose is based on the algorithm by Mannila and Toivonen (1996). Similar algorithms have been proposed by Srikant and Agrawal (1995, 1996).

3.1 The Basic Algorithm

A number of basic concepts have to be defined before we can present the algorithm proper. The following definitions are based on [Mannila & Toivonen, 1996].

An *event sequence* is an ordered sequence of events. Each event is described by several attributes and an associated time of occurrence. Extensionally, an event corresponds to a row in the data set. In our database an event is described by the attributes **Actor1** (field *source code*), **Actor2** (*target code*) and **Action** (*event code*). The *date* field specifies the time of occurrence.

An *episode* is a combination of events with a partially specified order; episodes are the basic concepts from which event prediction rules will be derived. They are represented as conjunctive patterns of conditions over attributes. Figure 2 gives an example of a (fictitious) episode.

```
x.Actor1 = ISR &
x.Actor2 = USA &
ask-for-aid(x.Action) &
y.Actor1 = USA &
y.Actor2 = ISR &
assistance(y.Action)
```

Figure 2: Intensional definition of an episode

This episode describes situations where Israel asks the United States for aid, and the US (later) gives some assistance. *x* and *y* are event variables that represent individual events and implicitly specify the temporal order of the episode. *ask-for-aid* and *assistance* are predicates on actions that are true if an action belongs to a particular (user-defined) class. Generally, a *predicate* is a condition that can be true in one event or across two events. We consider two kinds of predicates in this context, unary (which apply to one event) and binary ones (which apply to one or two events) - see section 3.2.1.An episode *occurs* on an event sequence if there is a (not necessarily contiguous) sequence of events that satisfy the conditions in the episode. The set of occurrences thus constitutes the extension of the episode.

An occurrence of an episode *P* is *minimal* if *P* does not occur at any proper subinterval. The *set of minimal occurrences* of an episode *P* is denoted by *mo(P)*. The *size* of an episode is the number of predicates that are true in the episode. An episode is *simple* if it contains no binary predicates (see section 3.2.1 below). And finally, an episode is called *frequent* if the number of its minimal occurrences |*mo(P)*| is larger than a given frequency threshold *min-freq*.

The algorithm consists of two parts. It first finds all frequent episodes in the data, and then generates rules from these that satisfy certain confidence conditions.

3.1.1 Finding Frequent Episodes

The following algorithm, which finds all frequent episodes for a given frequency threshold, is taken directly from Mannila & Toivonen (1996).

The basic algorithm is very similar to the Apriori-Algorithm introduced in [Agrawal & Srikant, 1994]. In the first step we look for episodes of size 1 in the database, meaning that we are searching for episodes including one predicate. In the following iteration the algorithm always selects two 'joinable' episodes, creates a candidate episode with an incremented size from these, and computes the minimal occurrence of this new episode from the minimal occurrences of the two subepisodes. If the frequency of the resulting candidate episode is greater than or equal to the user-defined *min-freq*, the episode is retained. The issue of selecting the two sub-episodes P_1 and P_2 in line 10 is not entirely trivial; [Mannila & Toivonen, 1996] discuss this in a bit more detail. The loop continues until no more episodes with a frequency greater than or equal to the user-defined *min-freq* can be found.

<u>Algorithm 1</u>: Discovery of frequent simple serial episodes [Mannila & Toivonen, 1996]

Input: event sequence S
unary predicates Γ
binary predicates Δ
frequency threshold min-freq

Output: all frequent simple episodes from the set of
possible episodes $\varepsilon_S(\Gamma,\Delta)$
(and their minimal occurences)

```
1. $C_1$:= the set of episodes of size 1 in $\varepsilon_s(\Gamma,\Delta)$;
2. for all $P \in C_1$ compute mo(P);
3. $L_1$ :={$P \in C_1$ | P is frequent};
4. i := 1;
5. while $L_i \neq \emptyset$ do
6.     $C_{i+1}$:={$P$ | $P \in \varepsilon_s(\Gamma,\Delta)$, subepisodes of P are in $L_i$};
7.     i++;
8.     for all $P \in C_i$ do
9.         select subepisodes $P_1$ and $P_2$;
```

```
         compute mo(P) from mo($P_1$) and mo($P_2$);
   end;
   $L_i$ :={P $\in$ $C_i$ | P is frequent};
  end;
for all i and all P $\in$ $L_i$ do
   output P and mo(P);
end;
```

For our application we have made some little changes to the original algorithm (mainly to do with binary predicates and search schemata), but the basic structure has remained unchanged.

3.1.2 Generating Prediction Rules

Extracting rules from the found frequent episodes is straightforward. An *episode rule* is an expression $P[V] \Rightarrow Q[W]$, where P and Q are frequent, and V and W are real numbers indicating the maximum time that may pass between the start of the “antecedent episode” P and the end of P, and between the start of P and the end of the “consequent episode” Q, respectively. What we are interested in is all rules with a given *minimum confidence*, where the confidence is computed essentially as the ratio of the number of (minimal) occurrences of P followed by Q in the time window $start(P)+W$ vs. the number of minimal occurrences of P within $start(P)+V$. If the resulting value is at least as large as a user-specified minimal confidence *min-conf* we consider the rule as valid.

To illustrate, Figure 3 shows an episode rule. Here, the interpretation is that if Israel asks the United States for some kind of support and the United States promises support within 7 days, the support is given within another 7 days.

```
x.Actor1 = ISR &
x.Actor2 = USA &
ask_for_aid(x.Action) &
y.Actor1 = USA &
y.Actor2 = ISR &
promise_aid(y.Action)
[7]
⇒
z.Actor1 = USA &
z.Actor2 = ISR &
assistance(z.Action)
[14]
  (confidence = 56,23 %, frequency =123)
```

```
ISR USA ask_for_aid
USA ISR promise_aid
=>
USA ISR assistance
  (C=0.56, F=123, W=14, V=7)
```

Figure 3: An episode rule in long and short-hand notation.

In the remainder of the paper, we will use the short-hand notation that is displayed in the lower part of figure 3.

3.2 Defining the Search Space

To allow the user to easily customize the representation language of the system and to guide and constrain the search in a problem-dependent way, we devised two quite straightforward extentions to the basic algorithm.

3.2.1 Unary and Binary Predicates

The first concerns the possibility for the user to define a hierarchy of predicates. Generally, a *predicate* is a condition on the attributes of an event, or a constraint on attributes of two different events. The former type is referred to as *unary predicates*. Unary attributes either test for equality with a constant, as in

```
Actor1=USA
```

or they test whether an attribute value belongs to a predefined class. That means that the user will be able to create his or her own taxonomy of predicates. For example, if the user has made the definition

```
arabic-country(X):
SYR
JOR
IRQ
 ....
```

then *arabic-country(x.Actor1)* will be true in an event *x* if the value of its attribute *Actor1* occurs in the definition of *arabic-country*. The user can specify a hierarchy of predicates.
Analogously, *binary predicates* allow testing for equality, for example

```
equal(x.Actor1,x.Actor2)
```

and again testing for the occurrence in a predefined set of constants like

```
neighbors(x.Actor1,x.Actor2),
```

where the relation neighbors might be defined as

```
neighbors(X,Y):
IRQ, IRN
ISR, JOR
ISR, SYR
…..
```

3.2.2 Dependency Schemata

To allow for a focused, problem-specific search on the data, we created a simple language that allows the user to specify conditions on the patterns he wants to search for. An example of such a schema is

```
Actor1=USA, Actor2=ISR
Actor1=ISR, Actor2=USA

ask-for-aid( Action)
assistance( Action)
promise-aid( Action)
```

where we search for rules that involve the United States and Israel (as either active or passive actor) and that involve some kind of help action (where *ask-for-aid* etc. are again user-defined unary predicates on actions).
Another example for a search schema including the search for binary predicates is

```
neighbors(Actor1,Actor2)

threat(Action)
protest(Action)
Action=223
```

where we are searching for threats and protests (user-defined abstract predicates) and the specific action 223 between neighbors. 223 is the KEDS event code for a military engagement.

4 Experimental Results

4.1 Database Characteristics

As mentioned in section 2, we were using two databases, one that originated from the lead texts of REUTERS newswire articles, and one that results from the entire text of the same articles. From the latter, we constructed a third set that consisted of the entries that concerned the second gulf war (Iraq’s occupation of Kuwait and USA’s intervention) and its consequences. For this set, we simply used the last 1022 days of the *full* dataset, so that we included all events from the crisis itself, as well as all subsequent events up to the end of the database.

	Lead	***Full***	***Gulf***
No. Events	57,131	304,402	57,195
No. Unique Daily Events	54,466	265,273	46,267
No. Different Events	17,123	49,550	14,438
No. Days	7,291	7,291	1022
Avg. Unique Events/Day	7.47	36.38	45.27
No. Actor1	164	199	157
No. Actor2	166	196	165
No. Action	65	66	64

Table 1: Characteristics of the used datasets.

Table 1 shows some characteristics of these three datasets. The *full* set contains 304,402 events. In these about 200 different actors appear in the active or passive actor fields, 66 different actions appear in the events (which means that about half of the 110 events of the WEIS coding do not appear at all). On the average, each day contains 36 events. In the subset that is concerned with the gulf war, each day contains about 45 events, indicating that this is a period of increased activity. The *lead* dataset contains much fewer events, which reflects its encoding for which not the entire newswire articles were used.

The *gulf* data set has about the same number of examples as the *Lead* dataset. However, they differ in some of their other characteristics, in particular in the number of *unique daily events* (counting identical events only once per day) and the number of *different events* (counting identical events only once for the entire dataset). The reason for this is that the events from the *gulf* data set originate from fewer newswire articles, which are described in more detail and are thus more repetitive (for example through referral to recent past events). All subsequent experiments were performed on sets with unique daily events.

The four most frequent actions in the *full* database are 023 (neutral comment), 031 (meeting), 223 (military engagement), 032 (visit), each of which occurs more than 20,000 times, and more than 4,000 times in the *lead* database. All other types of actions occur significantly less frequently. The most frequent actors are the USA, Iran, Iraq, followed by the United Nations, and Israel. It is interesting to note that while USA is the most frequent first, active actor by a wide margin, Iraq is the most frequent second, passive actor by an equally wide margin (both occur more than 50,000 times in the *full* set, followed by the other of the two and Iran with slightly more than 30,000 occurrences). The most frequent episodes are military engagements betwenn Iran and Iraq, Iraq and Iran, USA and Iraq, and Israel and Lebanon, followed by neutral comments between these parties, and the United Nations' visits to Iraq.

4.2 Quantitative Results

In a first experiment, we tried to find associations between events. We analyzed the *lead* and *gulf* datasets with the same set of parameters: The time window was chosen to incorporate 7 days (W=7 and V=7), we looked for episodes of length up to 4, and we used a set of 17 unary predicates that encode abstractions of the action fields. For the *lead* and *gulf* data sets, we set *min-freq (mF)* to 20 and *min-conf (mC)* to 0.5. For the data set that contained all examples generated from the *full* texts we had to use more restrictive values (mC=0.7, mF=100, V=W=5). We also give the number of rules found in the other datasets using these parameters to allow a rough extrapolation.

	Lead	*Gulf*	*Full*

mF=20, mC=0.5
V=7, W=7			

No. Episodes	8,073	167,993	n.a.
No. Rules	2,307	104,449	n.a.

mF=100, mC=0.7
V=5, W=5			

No. Episodes	94	153	12,146
No. Rules	0	26	1,746

Table 2: Number of found episodes and rules (events with 17 abstract action predicates).

The difference between the *lead* and the *gulf* datasets in terms of the number of episodes and sequences found is striking. Although both sets are of similar size, the *gulf* set seems to be much more repetitive (even though duplicate events that occurred on the same day had already been removed). This, of course, is again explained by the lack of detail in the encoding. Table 2 clearly illustrates the effect of increased repetitiveness on the complexity of the task.

Figure 4 shows the number of found rules – on a logarithmic scale - found for various settings of the minimum confidence and minimum frequency parameters for the *gulf* data set. Apparently, the number of rules found grows doubly exponentially with decreasing minimum frequency. Lowering minimum confidence, on the other hand, results in a sub-exponential growth of the number of found rules. A look at the original (non-logarithmic) scale (not shown) shows that the growth is still super-linear. Note that the growth along the minimum frequency axis (for constant minimum confidence) roughly corresponds to the growth in the number of episodes found and thus to the time complexity of the task. We repeated the same experiment for the *lead* data set, where the shape of the mesh looked almost identical, although the absolute numbers of found rules were lower by two orders of magnitude (cf. table 2).

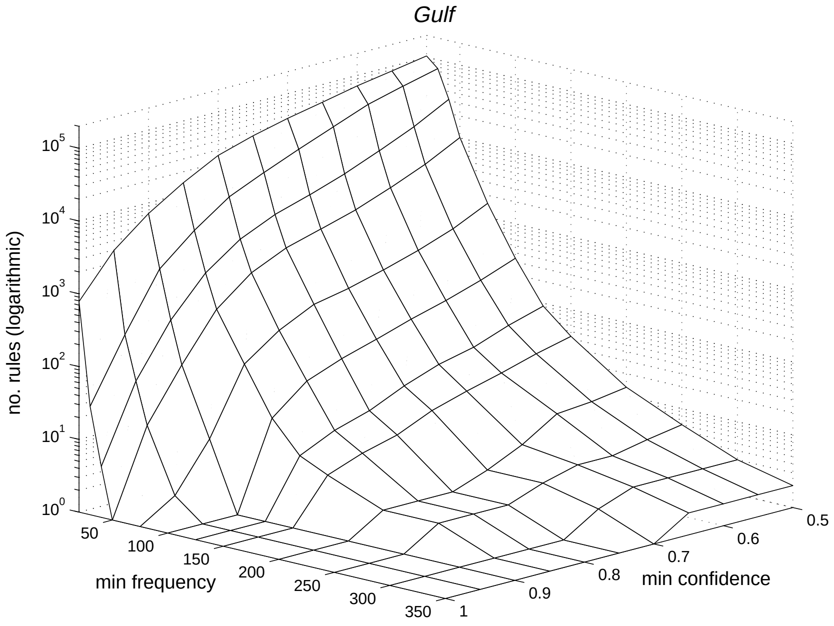

Figure 4: Number of rules found in the *gulf* set for various settings of the mininum confidence and minimum frequency parameters (logarithmic *z* axis).

4.3 Anecdotal Results

In this section, we show some of the rules that resulted from our experiments. In general, finding interesting rules in the large amounts of rules that are generated is a data mining task for itself. The rules with highest confidence and highest support are typically not very interesting. For example, the rules with the highest confidence and - among those - the highest frequency in the *gulf* set is the rule

```
USA IRQ demo
IRQ USA 223
USA IRQ 223
=>
USA IRQ 223
```

with a confidence of 1.0 and a frequency of 45 (`demo` is an abstract predicate that comprises military and non-military demonstrations and 223 is the event code for military engagement). Subsequent rules are of a similar nature. For example, the rule

```
USA IRQ comment
RUS IRQ comment
UNO IRQ comment
=>
UNO IRQ comment
```

occurs 38 times, also with a confidence of 1, which seems to represent a typical communicative sequence. Repeated conditions, like in the previous rules, happen quite frequently. For example, the three most frequent rules are all based on a four-fold repetition of the event `USA IRQ comment`. The three rules are obtained by placing implication sign '=>' after the first (C=0.66, F=381), second (C=0.75, F=337), or third (C=0.87, F=289) condition. The last case, for example, means that whenever the USA has commented three times within 7 days on the IRQ there is a 87% chance that a fourth comment happened later in the same time window.

Clearly, such rules are of limited interest. In order to aid us in the task of detecting more interesting rules, we wrote a variety of post-processing filters that allowed us to efficiently sift through the output of our rule finder, to sort the rules according to various criteria, and to extract those that match specified patterns on the right-hand side or left-hand side of the rule.
For example, filtering out all rules in which the same condition occurs more than once reduces the number of found rules for the parameter setting in the upper half of table 2 from 104,449 to 38,393. Among these, the most frequent is

```
UNO IRQ meet
IRQ UNO meet
IRQ UNO protest
=>
UNO IRQ comment
   (C: 0.63, F:79)
```

We have also tried to focus on more interesting subsets. For example, filtering for rules that predict a threat gives the following most frequent rule:

```
USA IRQ comment
IRQ UNO protest
UNO IRQ comment
=>
USA IRQ threat
   (C: 0.56, F: 70)
```

A rule with confidence 1 for predicting a military action (223) of the USA against the Iraq is, for example:

```
IRQ UNO diplomatic restriction
UNK IRQ 223
=>
USA IRQ 223
  (C: 1.0, F: 20)
```

We have also tried to analyze particular questions. Kovar (1999) shows several rules detected in the first gulf war (IRN/IRQ), the second gulf war (USA/IRQ), the relations between USA and Israel, Israel and Palestinian organizations, Israel and Lebanon, and Israel and its Arabic neighbors. Each question was investigated with a variety of parameter settings. The procedure was as follows: First we started to look for rules between a limited set of participants. This can be done with a relatively low frequency. Then, we gradually increased the minimum frequency and added additional unary predicates to the background knowledge (adding unary predicates can be seen as a step for generalizing the found rules, hence the frequency of the rules can be expected to go up). The window size parameters W and V were also varied. The results ranged from simple action sequences (a military engagement following another military engagement within two days with a confidence of 95.26%, and within 20 days with a confidence of 99.99%), to more specific rules that seem to constitute characteristic (though deplorable) action patterns, as in

```
LEB ISR violation
LEB ISR violation
 =>
ISR LEB 223
   (C: 0.61, F: >60, V=10, W=20)
```

or

```
USA ISR protest
ISR LEB 223
 =>
ISR LEB 223
   (C: 0.77, F: >60, V=10, W=20)
```

4.4 Efficiency and Pre-Processing Issues

Although the Apriori algorithm and its derivatives are typically very fast, the size of the current database allowed us to study only subsets of the dataset. In order to answer interesting questions like

“What events imply a military engagement between A and B within the following n days?”

we basically have to generate all possible episodes for the interesting time window, and then filter this set for occurrences that involve “A B 223” on the right-hand side. Likewise, for analyzing the consequences of certain events, we have to filter the results (of a complete search) for occurrences of the events in question in the left-hand side of a rule.

Desirable would be a filter that is able to effectively constrain the search for such types of problems. This, however, is not possible because such a constraint should only apply to one event of an episode and leave all degrees of freedom to the rest. We have experimented with straightforward pre-processing techniques that reduce the number of examples for such types of problems.

For example, if we are interested in events that predict a military engagement between Israel and Lebanon within one week, we mark all occurrences of “ISR LEB 223” and remove all events that occurred more than 7 days before such an event. In this example, this procedure reduced the size of the dataset from 304,402 to 160,071, i.e. to almost half of the size of the original set. While the rule finder will only detect a fraction of the frequent episodes of the original data set (and thus be more efficient), all of the rules that are relevant to the subsequent filter query should still be present.

4.5 Binary Predicates

The biggest failure of the project so far was the use of binary predicates. We tried to find simple rules that involve an equality or a *neighbor* predicate (see section 3.2.1), but the process is too inefficient to be tractable in reasonable times for any of the three datasets we looked at. A fundamental problem was already noted by Mannila & Toivonen (1996). In the terminology of their paper, Lemma 7 does not hold for general episodes. The problem is that episodes that involve binary predicates cannot be detected solely with minimal episodes because an occurrence of the episode `equal(x.Actor1,y.Actor1)` may be nested within another occurrence (see Manilla & Toivonen for a more detailed example). In fact, we believe that the problem is more prevalent: In the simple event sequence A,B,B,C (A, B and C represent events) the minimal occurrences of the episodes A,B and B,C do not overlap. Hence, the episode A,B,C cannot be detected using a simple join of the episodes A,B and B,C. We simply ignored this problem in both cases, hoping that this sacrifice of completeness would buy us the necessary gain in efficiency.

However, the main reason for the failure to get usable results involving binary predicates seemed to be that computation of the matches for a binary predicate is computationally too expensive for the task at hand. Imagine, for example, that we look for frequent episodes in a time window of 7 days. In the simplest case, the *lead* dataset, we have about 7.5 events per day, i.e. the window encompasses more than 50 events. All possible combinations of these events have to be examined to determine that a binary predicate does not occur in this window[1]. Hence the effort is quadratic in the number of events in the window. Note that it is not possible to use information obtained from the unary predicates to restrict the search space. The predicate `equal(x.Action,y.Action)` may be frequent although none of its instantiations (concrete values for the action) may be frequent.

Our conclusion from this is that the use of binary predicates is intractable for applications with larger time windows such as ours. In future work, we will either have to develop more sophisticated, possibly heuristic search techniques to cope with this problem, or alternatively, apply more intelligent pre-processing techniques that allow us – in co-operation with a domain expert – to reduce the number of events per day by dynamically filtering out events that are irrelevant to the particular questions.

5 Related Work

Oates et al. (1998) briefly describe some anecdotal results of applying their MEDD (Multi-Event Dependency Detection) algorithm to the KEDS data. However, they only used a tiny subset of 886 events, and they apparently did not go on to investigate the KEDS database in any more detail. Our experiments seem to be the first large-scale attempt at applying symbolic data mining algorithms to the KEDS database in its entirety.

Schrodt (1997) uses Hidden Markov Models to detect early warning indicators (3 to 6

[1] Determining that it occurs may be a little faster, in particular if one is only interested in computing minimal occurrences.

months lead) for Tit-for-Tat (TFT) conflicts between Israel and Lebanon, i.e. the appearance of "`ISR LEB 22x`" followed by a counter-action "`LEB ISR 22x`" within two days. LEB typically stands for PLO (before 1982) or for Amal or Hizbollah militia (after 1985).

Our methods have the potential of producing results that could be interpreted as early warning indicators. For example, with the application of the pre-processing technique briefly described in section 4.4, we discovered several patterns that were indicative for a military action of Israel against Lebanon. The more interesting rules in this case look like this:

```
LEB ISR 223
ISR LEB threat
=>
ISR LEB 223
  (C: 0.79, F:77)
```

Violent actions of Lebanese groups against Israel that are followed by a threat of Israel, typically (79% of the time) result in an Israeli counter-action within 7 days (the window size that we chose for this analysis). However, once more, the main deficiency is the lack of efficient means for focussing the search on the interesting parts, which makes big searches that would incorporate windows of, say, 30 days or more, infeasible for the entire database.

6 Discussion

We view our work as a first study that delivered first results but mainly focussed on identifying the problems that have to be tackled in future work.

The main weakness of this study is a certain lack of focus with respect to the political results. This was caused by two major factors: the abundance of found rules (the vast majority of which were not interesting) and the lack of input from a domain expert.

From a technical point of view, our main result is the impracticality of the straightforward approach for incorporating binary predicates. Mannila & Toivonen (1996) have predicted this problem but even our simplistic approach of sacrificing completeness for a gain in efficiency did not yield satisfying results. We see this as a main topic for further research.

From a political science point of view, we should stress that we could only give a fraction of the rules that looked interesting to us. Kovar (1999) describes many more experiments, and there are innumerable other questions that we have not investigated. Also, we are laymen in political science. These results should be discussed with political scientists, or, better, the program should be extended so that an unskilled user can interactively query the database for interesting patterns, preferably with a visual query interface. Currently, we are trying to contact political scientists for helping us to focus on important research questions on this and related problems.

In any case, our results have demonstrated that a fully automated approach to detecting interesting episodes or rules, for example with the goal of early warning, has its limitations on the databases we analyzed. At least in this domain, high frequency and high confidence are not necessarily correlated with interestingness. Additional domain knowledge is needed for focussing the search, and additional algorithmic research is needed for doing this efficiently.

Acknowledgments

This research was sponsored by the Austrian National Bank (OeNB) under grant number 7295. Financial support for the Austrian Research Institute for Artificial Intelligence is provided by the Austrian Federal Ministry for Science and Transport (BMWV).

References

[Agrawal & Srikant, 1994] Agrawal, R. and Srikant, R. Fast Algorithms for Mining Association Rules in Large Databases. *Proceedings of the Twentieth International Conference on Very Large Data Bases* (VLDB’94), 487-499.

[Agrawal & Srikant, 1995] Agrawal, R. and Srikant, R. Mining Sequential Patterns. *Proceedings of the Eleventh International Conference on Data Engineering* (ICDE '95), 3-14, Taipei, Taiwan, March 1995.

[Bercovitch & Langley, 1993] Bercovitch, J. and Langley, J. The Nature of Dispute and the Effectiveness of International Mediation. *Journal of Conflict Resolution 37(4)*, 679-691, 1993.

[Fürnkranz et al., 1997] Fünkranz, J., Petrak, J., and Trappl, R. Knowledge Discovery in International Conflict Databases. *Applied Artificial Intelligence 11(2)*, 91-118.

[Gerner & Schrodt, 1996] Gerner, D.J. and Schrodt, P.A. The Kansas Event Data System: A Beginner's Guide with an Application to the Study of Media Fatigue in the Palestinian Intifada.1996. http://ravens.ukans.cc.edu/~keds

[Kovar, 1999] Kovar, K. *Data Mining in Ereignisfolgen – Experimente mit der KEDS-Datenbank.* Master’s Thesis. Department of Medical Cybernetics and Artificial Intelligence, University of Vienna, September 1999. In German.

[Mannila et al., 1995] Mannila, H., Toivonen, H., and Verkamo, A.I. Discovery of Frequent Episodes in Event Sequences. *Proceedings of the First International Conference on Knowledge Discovery and Data Mining (KDD’95),* 210-215.

[Mannila & Toivonen, 1996] Mannila, H. and Toivonen, H. Discovering Generalized Episodes Using Minimal Occurrences. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD’96).*

[Oates & Cohen, 1996] Oates, T. and Cohen, P.R. Searching for Structure in Multiple Streams of Data. *Proceedings of the Thirteenth International Conference on Machine Learning,* 346 - 354. 1996.

[Oates et al., 1998] Oates, T., Jensen, D. and Cohen, P.R. Discovering Rules for Clustering and Predicting Asynchronous Events. In *Predicting the Future: AI Approaches to Time Series Workshop, AAAI-98,* 73-79. 1998.

[Pfetsch & Billing, 1994] Pfetsch, F.R. and Billing, P. *Datenhandbuch nationaler und internationaler Konflikte.* Nomos Verlagsgesellschaft, Baden-Baden, 1994.

[Schrodt, 1997] Schrodt, P.A. Early Warning of Conflict in Southern Lebanon using Hidden Markov Models. Paper presented at the annual meeting of the *American Political Science Association,* August 1997, Washington DC.

[Schrodt & Gerner, 1998] Schrodt, P.A. and Gerner, D.J. An Event Data Set for the Arabian/Persian Gulf Region 1979-1997. Paper presented at the 1998 annual meeting of the

International Studies Association, March 1998, Minneapolis.

[Sherman, 1988] Sherman, F.L. Sherfacs: A New Cross-paradigm, International Conflict Dataset. Paper presented at the 1988 annual meeting of the *International Studies Association*, 1988.

[Srikant & Agrawal, 1995] Srikant, R. and Agrawal, R. Mining Generalized Association Rules. *Prodeedings of the 21st VLDB Conference*, Zurich, Swizerland,1995.

[Srikant & Agrawal, 1996] Srikant, R. and Agrawal, R. Mining Sequential Patterns: Generalizations and Performance Improvements. *International Conference on Extending Database Technology* (EDBT '1996).

[Trappl et al., 1996] Trappl, R., Fürnkranz, J., and Petrak, J. Digging for Peace: Using Machine Learning Methods for Assessing International Conflict Databases. In *Proceedings of the 12th European Conference on Artificial Intelligence (ECAI-96)*, Budapest, 1996.