Forecasting Conflicts using N-gram Models

Alireza Bakhtiari, Camille Besse & Luc Lamontagne
Department of Computer Science
Laval University, Quebec, Qc, G1V 0A6

1 Introduction

Analysing international political behaviour based on similar precedent circumstances is one of the basic cognitive devices that policymakers use to define and evaluate current situations. These analyses are based on international interactions and events that occur between political actors in different periods of history. In quantitative researches on international relations, these political behaviours are coded in terms of *event data* which, in simple terms, describe "who did (when/where) what to whom" [1]. Formally, event data are generated by examining newspaper headlines on a daily basis (or up to several times a day when there are many important altercations), determining the two main actors involved in that event, and assigning a numerical code for the type of interaction between them.

In the past, few researchers used machine learning methods to predict and monitor conflicts around the world. Trappl [12] regroups a handful of the last ten years of research in the interdisciplinary field of machine learning for peace. Two chapters from this book are close to the case we studied. Trappl himself used case based reasoning in order to learn decision trees for predicting the outcome of future conflict mediation attempts [11]. Schrodt also worked using decision trees and neural networks on conflict databases [7, 8], but mainly used hidden Markov models to predict and monitor conflicts using automatically analysed news reports [6].

In this paper, we will evaluate the performance of N-gram models on the problem of forecasting political conflicts from sequences of events. For the current phase of the project, we focused on event data collected from the Balkans war in the 1990's. We begin by giving an overview of the corpus and the coding schemes used in Section 2, followed by our methodology and a discussion on the results obtained in Sections 3 and 4, respectively.

2 Pretreatment of Conflict Data Sets

Data sets we made use of were automatically extracted event series from news reports using the Kansas Event Data System (KEDS) project [9, 5][1]. These data sets are series of events formalized as pairwise interactions involving two participants, one acting against the other. Formally, a conflict is described as an event sequence $e_1, ..., e_n$ of tuples $e_i = \langle t_i, s_i, o_i, c_i \rangle$. where:

- t_i is the time-stamp (a number representing the date, by the day);
- s_i is the subject (the source of the action);
- o_i is the object (the target of the action);
- c_i is the code (the event/action type).

The categories of events are given according to the World Event/Interaction Survey (WEIS) [4] which roughly assigns higher codes to more hostile events. Events are distributed in 22 categories, inside of which they may be clustered into other subcategories. For sake of simplicity, we only kept the 22 main event types and compared them to the 4 event types that Schrodt used in [6]. This

[1] Data sets are available at http://web.ku.edu/~keds/data.html

latter categorization clusters actions into highly cooperative, mildly cooperative, mildly hostile and highly hostile classes. Table 1 shows an excerpt of the dataset after simplifying the event codes and adding the 4 event types, where the first column shows the time-stamp, the second represents the object, the third represents the subject, the fourth and fifth columns show the WEIS code and the simplified WEIS code, and the last column is the 4 event type code. Note that we also removed the participants involved in less than one hundred events, and denoted them by '—', meaning that the other participant may be anybody else.

32551	CRO	UNO	42	4	1
32552	BOSSER	MOS	150	15	3
32554	KSV	—	95	9	1
32556	BOS	USA	41	4	1

Table 1: Event dataset samples

In order to define the vocabulary needed by N-gram models, we chose to compare two types of event encoding schemes. In the first one, introduced by Schrodt in [6], we only consider a single actor at a time. This means that each event is split into two separate events; one considering the subject acting upon anybody, the other considering the object "being acted upon" by anybody. For instance, the event `32551 CRO UNO 42 4 1` is divided into the events `31138 CRO ---42 4 1` and `31138 ---UNO 42 4 1`. As a result, there are two codes for each actor, one when he is the subject, and one when he is the object. This type of encoding, which we named the *low-interaction scheme* (LO), roughly considers how a specific actor interacts with everybody. The other type of coding scheme simply considers all possible pairs of interactions, thus requiring much more codes but representing the possible interactions in a better way. We refer to it as the *high interaction scheme* (HI).

Therefore, a different code was assigned for all interactions of each actor and for each event type. For example, for the Balkans dataset where the top 11 actors were considered, the low interaction scheme with 4 event types results in 88 codes, while the high interaction scheme with 22 event types gives us 2904 codes. We also varied the number of actors and created datasets for 4, 6, 11, 50 and 99 actors, each of which were chosen according to their level of involvement in the conflict. Overall, results with 11 actors seem to present a good tradeoff between involvement and relevance of actors in the Balkans conflict.

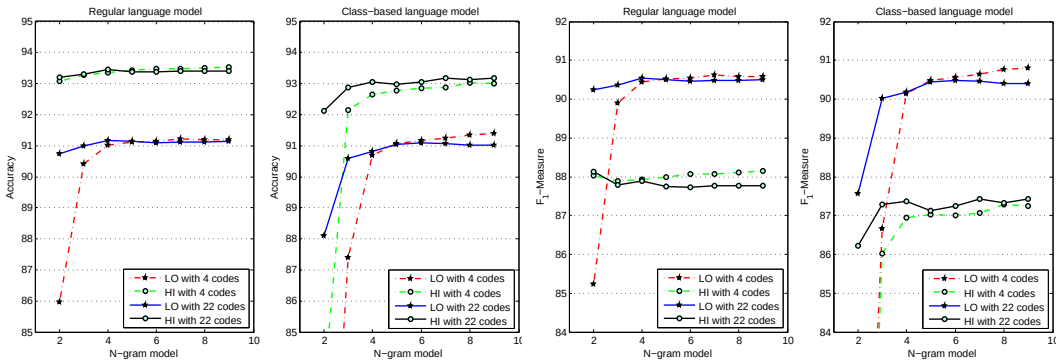
Finally, following the approach Schrodt used in [6], a week was defined as high conflict if the number of events with WEIS code above 18 was greater than a given threshold. This threshold was set to 20 force events in our experiments. We then took the one hundred events that happened one, three and six months prior to the start of that week in order to train and test our N-gram models[2]. As a result, we had six datasets of event sequences for every actor-coding scheme combination; one for high conflict weeks and one for low conflict weeks, for each of the three forecast periods.

3 Methodology

We used N-gram language models to learn sequences of recurrent patterns in the event database. Formally speaking, a *language model* is defined as a probability distribution $p(s)$ over a sequence of words s that reflects how frequent the sequence is within the whole text. In our context, the sentences that make up the dataset are sequences of 100 consecutive events based on the coding scheme chosen. Therefore, for the events in Table 1, one possible sequence could be `281 171 353 121` where each word (code) in the sequence is the code representing the event (in this case, based on a high interaction scheme with 4 event codes, and considering the top 11 actors involved).

Separate event sequences were prepared for each different actor set (i.e. sets containing the top 4, 6, 11, 50, or 99 actors) and for every coding scheme (high interactions or low interactions, with 4 or 22 possible event codes). For every actor-coding scheme combination, we built different event datasets for groups of sentences leading to high conflict or low conflict weeks, with 1 month, 3 months and 6 months forecast periods, a total of 120 event datasets. Each of the event datasets were further divided into training and test sets based on a 5-fold cross validation procedure with random selection.

[2]Why 20? why 100? These are free parameters and explanations are given in [6].



(a) Accuracy (b) F_1-measure

Figure 1: Results for the top 11 actors with a 28 day forecast period.

Using the event datasets, N-gram models were estimated for all of the possible combinations described above using the SRI Language Modeling toolkit (SRILM) [10]. In each case, for any actor-coding scheme-forecast period chosen, separate language models were trained over event datasets corresponding to high conflict and low conflict weeks, and the Witten-Bell discounting method was used to smooth the probability distributions. A binary classifier was then used to label the event sequences; in other words, every sentence in the test set was evaluated using both the high conflict and low conflict language models, and the perplexity of each model was calculated counting all input tokens. Consequently, sentences (sequences of 100 consecutive events) were individually classified as leading to a high or low conflict week based on which model gave lower perplexity to that sentence.

In the next phase of the project, we tried to model events not as interactions among two single actors, but as different coalitions acting against one another. Following the approach in [1], actors were attributed to groups in a way that intra-group negative interactions were rare or nonexistent, while inter-group conflicts were frequent and serious. For instance, two of the main actors in the Balkans war, namely Serbia and Serbs in Bosnia (represented by SER and BOSSER, respectively) appear to have shared a common strategic outlook throughout the conflict, and it seems natural to assign them to a single group. With these criteria in mind, we identified four groups of actors which had quite similar strategic views during the conflict period using the results in [1]. These consist of the two main groups in conflict; namely {BOS(Bosnia)} against {SER and BOSSER}, a mediator group {UNO and NAT (NATO)} along with an international group consisting of all other actors[3].

Using these groupings, we replaced actors with the corresponding group in which they belonged to, and prepared training and test sets for configurations similar to the ones described for regular N-grams. The IBM class-based N-gram model [2] was then applied to build the language models, which (for the simple bigram case) estimates sequences as:

$$p(w_i|w_{i-1}) \approx p(c_i|c_{i-1})p(w_i|c_i) \tag{1}$$

As with the regular N-gram approach, the language models were then used to find the perplexity of individual sentences in the test set, and high conflict or low conflict labels were assigned by comparing the perplexities given by each model.

4 Results

To compare the performance of our models, we computed various correctness measures for the different configurations described above. These measures are essentially based on the total number of sentences predicted as high conflict when we actually have a high conflict week (TP) or a low conflict week (FP), and the total number of sentences predicted as low conflict when we actually have a low conflict week (TN) or a high conflict week (FN). Accordingly, the performance measures

[3] A separate grouping scheme had to be applied for the 4 actor set, which was originally borrowed from Shrodt [6] and did not contain some of the most involved actors (i.e. UN and NATO, and BOSSER that was merged into SER)

	28 days	91 days	184 days
LO scheme	91.16	89.91	89.79
HI scheme	92.84	94.31	94.01

(a) Accuracy

	type	28 days	91 days	184 days
LO	high-conflict	90.56	86.15	86.14
	low-conflict	91.70	92.01	91.92
HI	high-conflict	87.01	77.45	76.21
	low-conflict	95.07	96.74	96.58

(b) F_1-measure

Table 2: Results of class-based 6-gram model for the top 11 actors.

calculated were the overall accuracy of each model (the relative number of correctly predicted high and low conflict weeks; i.e. $\frac{\text{TP+TN}}{\text{TP+FP+TN+FN}}$), true-positive and true-negative precisions (how many of the weeks predicted as high/low conflict were actually a high/low conflict week; i.e. $\frac{\text{TP}}{\text{TP+FP}}$ and $\frac{\text{TN}}{\text{TN+FN}}$), true-positive and true-negative recalls (of all actual high/low conflict weeks, how many were correctly labelled as high/low conflict; i.e. $\frac{\text{TP}}{\text{TP+FN}}$ and $\frac{\text{TN}}{\text{TN+FP}}$), and the F_β-measure for $\beta \in \{0.5, 1, 2\}$ that combines precision and recall to put more emphasis on either one of them.

Due to the large number of performance results obtained for all configurations, only the accuracy and true-positive F_1-measure (where precision and recall are evenly weighted) for 11 actors with a 28 day forecast period will be discussed here. Figures 1a and 1b show how each of our different coding schemes perform when evaluating N-grams for $N \in \{2, \ldots, 9\}$. While the overall performance of N-grams is impressive, there is little improvement in both the accuracy and F_1-measures for $N \geq 6$. Therefore, for this specific actor-coding scheme, it appears that sequences larger than 6 events in length don't carry much more information contributing to our model's performance[4]. Furthermore, both figures show that class-based N-grams perform as well as regular N-grams (or even do better); a behaviour which is also observed in other actor sets containing the top 4, 6, 50 and 99 actors. This inspiring result suggests that instead of using a large number of actors, we can achieve the same level of performance with considerably less codes (for example, 80 codes instead of 39600 codes by grouping the top 99 actors into 4 blocks in the HI scheme with 4 event codes).

The complete accuracy and F_1-measure results for the class-based 6-gram model are shown in Table 2. We can observe from this table that, in general, low interaction coding schemes (LO) are much better at correctly forecasting low conflict weeks (more TN's), and high interaction schemes (HI) outperform LO schemes when it comes to forecasting high conflict weeks (more TP's). This results in LO schemes having higher precision and recall with regard to high conflict weeks, which explains the gap between LO and HI schemes in Figure 1b. On the other hand, since the data is strongly skewed towards low conflict weeks (around 80% of the whole dataset) the number of TN's is larger than the number of TP's, and thus TN's will have more influence over the accuracy of the model. As a result, HI schemes have higher accuracy (Figure 1a), and higher true-negative F_1-measure compared to LO schemes.

5 Conclusion and Future Work

In this paper, we discussed the application of N-gram models to the problem of forecasting political conflicts. Our results show that these models have impressive results when applied to the Balkans war, with accuracies above 90% for most configurations. Analysis of the top frequent N-grams shows some interesting recurrent sequences of events, however, extracting meaningful patterns from the large number of data remains to be done in a future work. These models must also be used with other datasets to analyse their performance in forecasting more complicated conflicts with more involved actors, like the war in central Asia (between Afghanistan, Armenia-Azerbaijan and former Soviet republics). Furthermore, instead of manually assigning actors to groups, clustering algorithms could be used to determine the different groups of actors in conflict with each other.

[4]One might wonder why sequences of up to 6 consecutive events capture enough information to contribute to the final performance results. Looking at the top counts reveals many recurrent and interesting patterns in the data, however, discussion of these results is out of the limits of the current paper and further analysis is required in a future work.

References

[1] U. Brandes and J. Lerner. Visualization of conflict networks. *Nato Security Through Science Series E Human and Societal Dynamics*, 36:169, 2008.

[2] P.F. Brown, P.V. Desouza, R.L. Mercer, V.J.D. Pietra, and J.C. Lai. Class-based n-gram models of natural language. *Computational linguistics*, 18(4):467–479, 1992.

[3] C.A. McClelland. Event-interaction analysis in the setting of quantitative international relations research. *Photocopy. Los Angeles: Department of Political Science, University of Southern California*, 1967.

[4] Charles A. McClelland. *World Event/Interaction Survey Codebook*. (icpsr 5211), 1976.

[5] Shannon G. Davis Philip A. Schrodt and Judith L. Weddle. Political Science: KEDS, A Program for the Machine Coding of Event Data. *Social Science Computer Review*, 3(12):561–588, Fall 1994.

[6] Philip Schrodt. Forecasting conflict in the balkans using hidden markov models. In Robert Trappl and al., editors, *Programming for Peace*, volume 2 of *Advances in Group Decision and Negotiation*, pages 161–184. Springer Netherlands, 2006.

[7] Philip A. Schrodt. Predicting Interstate Conflict Outcomes Using a Bootstrapped ID3 Algorithm. *Political Analysis*, 2(1):31–56, 1990.

[8] Philip A. Schrodt. Prediction of Interstate Conflict Outcomes Using a Neural Network. *Social Science Computer Review*, 9(3):359–380, 1991.

[9] Philip A. Schrodt and Shannon G. Davis. Techniques and troubles in the machine coding of international event data. In *Proc. of the Meeting of the International Studies Association*, Washington, DC, 1994.

[10] A. Stolcke. Srilm-an extensible language modeling toolkit. In *Proceedings of the international conference on spoken language processing*, volume 2, pages 901–904. Citeseer, 2002.

[11] Robert Trappl, Johannes Fürnkranz, and Johann Petrak. Digging for Peace: Using Machine Learning Methods for Assessing International Conflict Databases. In *ECAI*, pages 453–457, 1996.

[12] Robet Trappl, editor. *Programming for Peace: Computer-Aided Methods for International Conflict Resolution and Prevention (Advances in Group Decision and Negotiation)*. Springer, 1 edition, February 2006.