The VOTE Program as Open Source

George V. Neville-Neil

April 26, 2024

1 Introduction

The VOTE system is a Common Lisp implementation of the work presented in the book Goal Based Decision Making by Stephen Slade https://www.amazon.com/Goal-based-Decision-Making-Interpersonal-Model/dp/0805813667. The code has evolved over several decades and is now available on github as a git repository. https://github.com/sslade/VOTE. At the time of this writing the system only contains information on the legislative bills, issues, and people of the United States in the late 1980s. Replacing the databases used by the system would allow it to be used in other political systems.

An example of how to run the current system is given in the README.md in the root of the repository.

The rest of this document contains a brief overview of the code, for those who wish to extend it, as well as a discussion of how this code can be maintained as an ongoing, open source, project.

2 Code Overview

All of the code for the system is contained under the lisp directory in the root of the repository. There remain some lingering vestiges of past code but we shall ignore those for now and only concentrate on the actively used components.

The pol/ directory contains the main pieces of code that handle the goal based reasoning described in the book listed in Section 1.

Each of the major sections has its own lisp file, shown in Table 1. The code is written in an object oriented (OO) style where each major component is given a class, and each class has defined methods. For example, the issue class is defined in the issue.lisp file, and contains all the code that is necessary to interact with the issues database as well as respond to queries sent to it. All of the major code sections follow this same pattern.

The langauge generator, which currently only handles English, is contained int he gen/ directory. It has a realtively good implementation of English grammar, suitable to the task making the output of issues, stances, bills and votes less taxing to the reader. Some amount of randomness is injected into the system so that each time a query is made of the system the output might be a bit different. The gen code has a test entry point that is useful in seeing how this works.

The data for the system is contained in Lisp textual databases found under the db/data/ directory. Whilst these databases can be manipulated directly there are also commands in each class (bill, issue etc.) that can be used to udpate the database entries.

File	Purpose
bill.lisp	Legislative bills, past and present
group.lisp	Special Interest Groups
issue.lisp	Societal and other issues that might go into a bill
member.lisp	Members of the legislature
stance.lisp	Handling the stances of members on issues

Table 1: Major Code Files

3 Ideas for Future Work

For the VOTE program to be a viable open source project it needs to have more engagement from people who find the concepts interesting, and want to apply them to other areas of goal based reasoning. The system is generic enough to allow for expansion in several different ways.

3.1 Langauge Translation

One of the suggested projects for students who worked with the system in the course at Yale, where this code originated, was to translate the system into another language. In fact there had been a French translation in earlier code but this was lost to the winds of time, and only a few vestiges remained. The code in the `gen/` directory is very much focused on English grammar and English political turns of phrase. Nonetheless it was possible to do a translation into Japanese, although without any updates to the political arena as data on Japanese bills would have been hard to come by. A fork of the code with Japanese was submitted as a project in 2024.

The current langauge code is not table driven, and the text and code are mixed together into each of the relevant classes (e.g. `noun.lisp`, `verb.lisp` etc.) A fully internationalized system would have to switch to tables for words and phrases, and different classes for each language, as the grammar is encoded in the langauge class, e.g. `english.lisp`, `japanese.lisp` etc.

A more full translation effort should be tied to creating a version of the databases so that they inhabit a different poltiical arena as discussed next.

3.2 New Political Arenas

The databases associated with the code only contain people, groups, issues and bills that relate to the US House of Representatives as of the late 1980s. The format of the data is clear and so importing other sets of data is straightforward, but getting access to that data may be a challenge. Political systems exist within independent langauge and cultural systems, which means that even trying to bringin databases that are nominally in English, for example from the UK Parliament, will require some amount of linguistic changes, as discussed in Section 3. Anyone who has read or heard poltiical discussions in both the US House and UK Parliament knows that the language used, while nominally English, differs in important respoects that change the meaning of words. Anything further afield from the US and the UK would, obiovusly, require an entirely new language translation of the `gen/` code.

3.3 New Areas of Reasoning

The `VOTE` system is a specific instance of goal based reasoning, one which handles roll call voting. The main base classes are capable of being adapted to other areas of goal based reasoning, by being abstracted away from the specific words used (issue, stance, bill etc.). A much longer term project would be to do this abstraction so that other types of systems could be tested against this type of reasoning.

4 Conclusion