

# An NLG System for Constituent Correspondence: Personality, Affect, and Alignment

William Kolkey, Jian Dong, and Greg Bybee

nScribe Technologies

{will, jian, greg}@nscribe.tech

## Abstract

Roughly 30% of congressional staffers in the United States report spending a “great deal” of time writing responses to constituent letters (Furnas, 2018). Letters often solicit an update on the status of legislation and a description of a congressman’s vote record or vote intention — structurable data that can be leveraged by a natural language generation (NLG) system to create a coherent letter response. This paper describes how PoliScribe, a pipeline-architected NLG platform, constructs personalized responses to constituents inquiring about legislation. Emphasis will be placed on adapting NLG methodologies to the political domain, which entails special attention to affect, discursive variety, and rhetorical strategies that align a speaker with their interlocutor, even in cases of policy disagreement.

## 1 Introduction

Recent work in the field of NLG has shifted from emphasizing the provision of information to aspects of tone and style. The shift follows from the increasing complexity of NLG systems themselves, which have evolved from vehicles of information delivery to sophisticated platforms aiming to persuade, engage, and even entertain. (Gatt and Krahmer, 2018). The trend holds particular relevance to the domain of political epistolography, in which qualities such as affect and personality are important for meeting the conventions of the genre.

This paper discusses these themes in the context of PoliScribe, an NLG system that is currently utilized by several dozen legislative offices across the United States, including federal and state representatives from California, Texas, and New York. To our knowledge, this is the first

instance of an NLG platform that has been applied to constituent communications.

## 2 System Overview

PoliScribe is designed to respond to constituents who are advocating for or against legislation. At its core, the platform follows a Leveltian model of text generation, whereby letters are constructed according to a rules-based schema consisting of content and document planning, sentence planning, and surface realization (Levelt, 1989; Reiter and Dale, 1999).

PoliScribe works by inputting a bill number into the user interface (e.g., *H.R. 5*) and selecting the member’s disposition toward the bill (e.g., *strongly support*, *undecided*, etc.). The system connects to public databases to identify the bill’s title, description, vote history, committee assignments, and sponsorship information. These and other informational elements are expressed within PoliScribe as independent sentences, noun phrases, adjectives, or subordinate clauses, and are realized through a process of aggregation with a bias for sentence variation.

The output of this operation is a letter that explains where a bill is in the legislative process and how the elected official feels about the legislation. The system is versatile, supporting more than a billion letter variations, depending on factors that include the stylistic preferences of the user, the legislative stage of the bill, and the relationship of the constituent’s policy views to that of their Representative’s.

## 3 Letter Generation

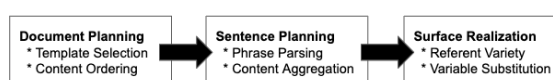


Figure 1: Schema for text generation.

PoliScribe uses a rule-based schema (outlined in Figure 1) to ensure a predictable and consistent output, essential qualities for any NLG system applied to a professional environment. Though rule-based, this approach does allow for a high degree of flexibility.

During the document planning phase, the system selects from one of several dozen templates that determine the ordering of informational elements in the letter. The selection of a template is informed by (a) the level of agreement between the Representative and constituent (b) the bill’s progress through the Legislature and (c) the stylistic preferences of the user, who can directly influence the properties of their letters through a *user settings* form on the system’s frontend.

To fully accommodate a wide variety of user preferences, templates are further partitioned into smaller blocks of informational elements, whose ordering is also influenced by selections made in user settings.

During the sentence planning phase, elements of a template are aggregated to create fluid and stylistically varied sentences. A demonstration of this functionality is a sentence that informs a constituent of the following facts: that a bill has passed a House of the Legislature, that it will be considered in a committee of the opposite House, and that it remains eligible for amendments.

Initially, those elements would be expressed within a template as independent variables: *Vote Outcome* + *Bill Location* + *Amendment Eligibility*. The system inspects these variables to discern their string values, which (for the sake of example) we will assume correspond with the following text: *The bill has passed {{ last vote location }} by a vote of {{ last vote }}*. + *The bill is on its way to {{ next vote location }}*. + *The bill is eligible for amendments*.

The system parses their grammatical structures to determine how different elements can be combined. The first two sentences are identified as sharing similar syntax (noun phrase + verb phrase + prepositional phrase) and a common subject (*the bill*). According to the internal rules of the system, it is permissible to combine these sentences with a coordinating conjunction and to delete the second occurrence of the noun subject. Hence: *The bill has passed {{ last vote location }} by a vote of {{ last vote }} and is on its way to {{ next vote location }}*.

The system then attempts to aggregate the newly generated sentence with the sentence related to amendment eligibility. Though these sentences share a grammatical subject, the internal rules of the system prevent aggregation as a compound sentence, since the first sentence now employs a coordinating conjunction. Instead, the template opts to treat the second sentence as a relative clause, a hard-coded solution for this particular language combination. The result is as follows: *The bill has passed {{ last vote location }} by a vote of {{ last vote }} and is on its way to {{ next vote location }}*, where the bill is eligible for amendments.

During the surface realization phase, content placeholders are replaced with real values and redundant referents substituted. The example from the previous paragraph would be realized as follows: *The bill has passed the floor of the House by a vote of 417 to 38 and is now on its way to the Senate Judiciary Committee, where it is eligible for amendments*.

#### 4 Navigating Affect and Style

How do you express support for legislation that is also supported by your constituent?  
 A. I am also supportive of this bill.  
 B. I share your support for this bill.  
 C. I will be voting for this bill.  
 D. None of the above. (Write your own text here.)

Figure 2: Onboarding question.

Since the objective of PoliScribe is to imitate the letter-writing style of a political officeholder, discursive features such as tone, personality, and affect are essential to engendering authenticity.

The system operates to imitate the stylistic tone of legislative offices by requiring users to fill out a language questionnaire prior to onboarding. An example of a typical onboarding question is given in Figure 2. Answers to questions influence both the system’s lexicon and rules for aggregation.

In addition, PoliScribe maintains a cognitive model for how the constituent might feel about a legislative development, termed *constituent satisfaction*, which influences the emotive content of the letter response. If a bill supported both by the constituent and their Representative is defeated on the House floor, for example, the system might use language expressing disappointment: e.g., *Despite my best efforts, the bill failed on the House floor*. If that same bill was instead approved by the House, the system might

say: *I am pleased to confirm the bill has passed the House of Representatives.*

## 5 Alignment Strategies

Just as important, PoliScribe takes into account the policy perspective of the constituent, so that letter-responses are framed around a shared vocabulary and political outlook. Such alignment strategies are common in dialogue and likewise pertain to epistolary correspondence, which are just another form of conversational act (Altman, 1982). They are also something that NLG systems are particularly well suited to perform, as shown by van der Lee's PASS system, which tailors summaries of football matches to audiences from opposing clubs (van der Lee et al., 2017).

Alignment is achieved by tagging bills with one of 220 topics appropriated from the Comparative Agendas Project (CAP), an organization that classifies legislation across democracies. Though CAP maintains a directory of coded bills, legislation can also be tagged independently through classification algorithms (Purpura and Hillard, 2006). PoliScribe next determines whether the bill is conservative, liberal, or bipartisan — primarily by considering the party affiliations of the bill's authors and cosponsors.

These efforts allow PoliScribe to employ issue-specific language that aligns with the policy perspective of the constituent. For example, a letter to a supporter of a *conservative* bill tagged *National Budget* might employ language emphasizing fiscal prudence. By contrast, a letter to a constituent supporting a *liberal* bill tagged *Labor Union* might employ language emphasizing fair labor practices. Issue-specific language can be edited and approved by user offices, ensuring that the system's rhetoric is consistent with their own policy outlook.

Tagging also enables PoliScribe to identify legislation that is related to bills that constituents write in about. This allows the system to educate the constituent about the Representative's vote record or emphasize areas of past agreement. Identifying related legislation is particularly useful for engendering a sense of alignment when the Representative disagrees with the constituent or has not yet taken a public position on a piece of pending legislation, but has voted for thematically similar legislation in the past.

For the sake of illustration, a letter generated by PoliScribe is produced below, with issue-specific language and related legislation emphasized in bold.

*Thank you for contacting my office regarding your support for AB 129, a bill related to plastic microfibers.*

*This bill was introduced by Assemblymember Bloom on December 4th, and would reduce the amount of plastic that enters our drinking water, by banning fabric that is more than 50 percent polyester. I share your support for this legislation, but should note that the bill has died in the Assembly's Environmental Safety and Toxic Materials Committee before reaching me for a vote.*

*Please know I am just as disappointed as you are with this result, and will keep your thoughts in mind as we continue to work on reforms that **promote healthy rivers and ensure a clean water supply**. If it is any consolation, you may be happy to know that I did vote for successful legislation last session **that would require all drinking water to be tested for plastic contamination**.*

*Thank you again for contacting my office, and please do keep in touch.*

## 6 Conclusions

PoliScribe has been in use by legislative offices since the beginning of 2019, with users reporting in informal surveys that the system has improved response times and has enabled more detailed replies to constituent letters. Essential has been the ability to accommodate the various stylistic preferences and affectations of individual elected officials, a level of customizability no doubt fundamental for any NLG system operating on behalf of distinct and forceful personalities.

## References

- Janet Gurkin Altman. 1982. *Epistolarity: Approaches to a Form*. Ohio State University Press, Columbus, OH.
- Alexander Furnas. 2018. [Legislative staff are spending an increasing mount of time on constituent services](#). LegBranch.org. Accessed: 2 July 2019.
- Albert Gatt and Emiel Kraemer. 2018. Survey of the State of the Art in Natural Language Generation: Core tasks, applications and evaluation. In *Journal of AI Research*, 61: 65-170.
- Willem J.M. Levelt. 1989. *Speaking: From Intention to Articulation*. MIT Press, Cambridge, MA.
- Stephen Purpura and Dustin Hillard. 2006. Automated Classification of Congressional Legislation. In *Proceedings of the 2006 International Conference on Digital Government Research*: 219-225.
- Ehud Reiter and Robert Dale. 1999. *Building natural language generation systems*. Cambridge University Press, New York, NY.
- Chris van der Lee, Emiel Kraemer, and Sander Wubben. 2017. Pass: A Dutch data-to-text system for soccer, targeted towards specific audiences. In *Proceedings of the 10th International Conference on Natural Language Generation*: 95–104.