Two-Stage Stochastic Natural Language Generation for Email Synthesis by Modeling Sender Style and Topic Structure

Yun-Nung Chen and Alexander I. Rudnicky
School of Computer Science, Carnegie Mellon University
5000 Forbes Ave., Pittsburgh, PA 15213-3891, USA
{yvchen, air}@cs.cmu.edu

Abstract

This paper describes a two-stage process for stochastic generation of email, in which the first stage structures the emails according to sender style and topic structure (high-level generation), and the second stage synthesizes text content based on the particulars of an email element and the goals of a given communication (surface-level realization). Synthesized emails were rated in a preliminary experiment. The results indicate that sender style can be detected. In addition we found that stochastic generation performs better if applied at the word level than at an original-sentence level ("template-based") in terms of email coherence, sentence fluency, naturalness, and preference.

1 Introduction

This paper focuses on generating language for the email domain, with the goal of producing mails that reflect sender style and the intent of the communication. Such a process might be used for the generation of common messages (for example a request for a meeting without direct intervention from the sender). It can also be used in situations where naturalistic email is needed for other applications. For instance, our email generator was developed to provide emails to be used as part of synthetic evidence of insider threats for purposes of training, prototyping, and evaluating anomaly detectors (Hershkop et al., 2011).

There are two approaches to natural language generation (NLG), one focuses on generating text using templates or rules (linguistic) methods, the another uses corpus-based statistical techniques. Oh and Rudnicky (2002) showed that stochastic generation benefits from two factors: 1) it takes advantage of the practical language of a domain expert instead of the developer and 2) it restates the problem in terms of classification and labeling, where expertise is not required for developing a rule-based generation system. They found that naive listeners found such utterances as acceptable as human-generated utterances. Belz (2005) also proposed a probabilistic NLG approach to make systems more robust and components more reusable, reducing manual corpus analysis.

However, most work usually focused on well-structured documents such as news and Wikipedia, while email messages differ from them, which reflect senders' style and are more spontaneous. Lampert et al. (2009) segmented email messages into zones, including sender zones, quoted conversation zones, and boilerplate zones. This paper only models the text in the sender zone, new content from the current sender. In the present work, we investigate the use of stochastic techniques for generation of a different class of communications and whether global structures can be convincingly created in the email domain.

A lot of NLG systems are applied in dialogue systems, some of which focus on topic modeling (Sauper and Barzilay, 2009; Barzilay and Lapata, 2008; Barzilay and Lee, 2004), proposing algorithms to balance local fit of information and global coherence. However, they seldom consider to model the speaker's characteristics. Gill et al. (2012) considered sentiment such as openness and neuroticism to specify characters for dialogue generation. In stead of modeling authors' attitudes, this paper proposes the first approach of synthesizing emails by modeling their writing patterns. Specifically we investigate whether stochastic techniques can be used to acceptably model longer texts and individual speaker characteristics in the emails, both of which may require higher cohesion to be acceptable.

2 Overview of Framework

Our proposed NLG approach has three steps: preprocessing training data, modeling sender style and topic structure for email organization, followed by surface realization, shown in Figure 1.

In preprocessing, we segment sentences for each email, and label email structural elements. This is used to create a structural label sequence for each email, and then used to model sender style and topic structure for email organization (1st stage in the figure). Content slots are also annotated for surface realization (2nd stage in the figure). Details are in Section 3.

From the annotated corpus, we build sender-specific and topic-specific structure language models based on structural label sequences, and use a mixture sender-topic-specific model to stochastically generate email structure in the first stage. The process is detailed in Section 4.

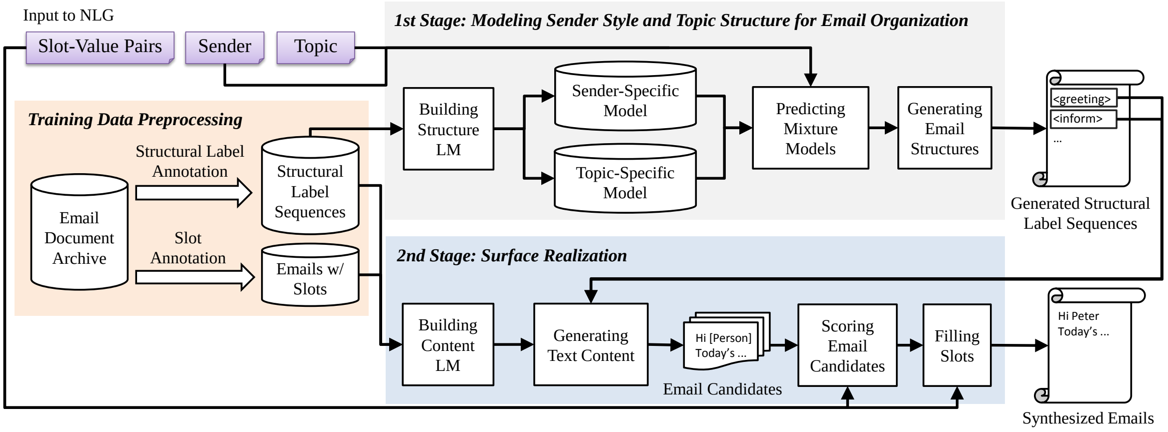


Figure 1: The proposed framework of two-stage NLG component.

In the second stage, we build a content language model for each structural element and then stochastically generate sentences using the sequence generated in the first stage. To ensure that required slot-value pairs occur in the text, candidates emails are filtered to retain only those texts that contain the desired content slots. These slots are then filled to produce the final result. Section 5 explains the process.

3 Training Data Preprocessing

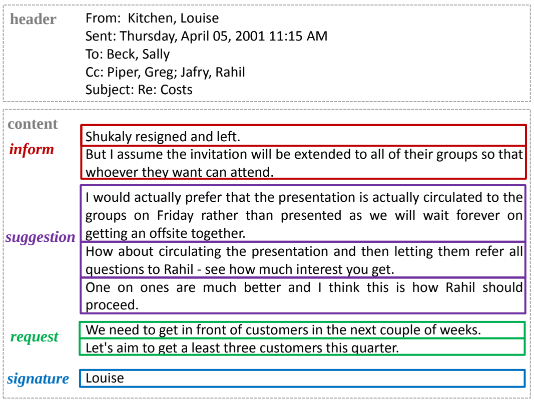
To model sender style and topic structure, we annotate the data with defined structural labels in Section 3.1, and data with slots to model text content of language in Section 3.2.

3.1 Structural Label Annotation

Based on examination of the corpus, we defined 10 email structure elements:

1. *greeting*: a friendly expression or respectful phrase, typically at the start of an email.
2. *inform*: to give or impart knowledge of a fact or circumstance.
3. *request*: the act of asking for something to be given or done, especially as a favor or courtesy.
4. *suggestion*: to mention or introduce (an idea, proposition, plan, etc.) for consideration or possible action.
5. *question*: an interrogative sentence in an form, requesting information in reply.
6. *answer*: a reply or response to a question, etc.
7. *regard*: to have or show respect or concern for, usually at the end of an email.
8. *acknowledgement*: to show or express appreciation or gratitude.
9. *sorry*: express regret, compunction, sympathy, pity, etc.
10. *signature*: a sender's name usually at the end of the email.

We perform sentence segmentation using punctuation and line-breaks and then manually tag each sentence with a structure label. We exclude the header of emails for labeling. Figure 2 shows an example email with structural labels.


header
From: Kitchen, Louise
Sent: Thursday, April 05, 2001 11:15 AM
To: Beck, Sally
Cc: Piper, Greg; Jafry, Rahil
Subject: Re: Costs

content

inform
Shukaly resigned and left.
But I assume the invitation will be extended to all of their groups so that whoever they want can attend.

suggestion
I would actually prefer that the presentation is actually circulated to the groups on Friday rather than presented as we will wait forever on getting an offsite together.
How about circulating the presentation and then letting them refer all questions to Rahil - see how much interest you get.
One on ones are much better and I think this is how Rahil should proceed.

request
We need to get in front of customers in the next couple of weeks.
Let's aim to get a least three customers this quarter.

signature
Louise

Figure 2: The email with structural labels.

3.2 Slot Annotation

The input to NLG may contain the information that needs to be included in the synthesized emails. Tokens in the corpus text corresponding to slots are replaced by slot (or concept) tokens prior to building content language models. Slots are classified into general class and topic class below.

3.2.1 General Class

We use existing named entity recognition (NER) tools for identifying general classes. Finkel et al. (2005) used CRF to label sequences of words in text that are names of things, such as person, organization, etc. There are three models trained on different data, which are a 4-class model trained for CoNLL[1], a 7-class model trained for MUC, and a 3-class model trained on both data sets for the intersection of those class sets below.

- 4-class: location, person, organization, misc
- 7-class: location, person, organization, time, money, percent, date

Considering that 3-class model performs higher accuracy and 7-class model provides better coverage, we take the union of outputs produced by 3-class and 7-class models and use the labels output by 3-class model if the two models give different results, since the 3-class model is trained on both data sets and provides better accuracy.

[1] http://www.cnts.ua.ac.be/conll2003/ner/

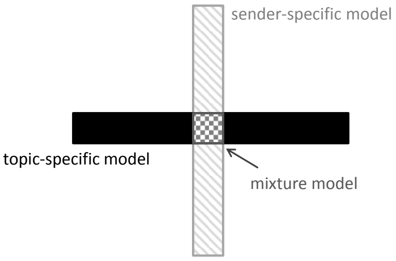


Figure 3: The visualization of the mixture model.

3.2.2 Topic Class

Many named entities cannot be recognized by a general NER, because they are topic-specific information. Accordingly we define additional entities that are part of the email domain.

4 Modeling Sender Style and Topic Structure for Email Organization

Given a target sender and topic focus specified in system input, email structures can be generated by predicted sender-topic-specific mixture models.

4.1 Building Structure Language Models

Based on the annotation of structural labels, each email can be expressed as a structural label sequence. Then we can train a sender-specific and a topic-specific structure model using the emails from each sender and the emails related to each topic respectively. Here the structure models are n-gram models with Good-Turing smoothing ($n = 3$) (Good, 1953).

4.2 Predicting Mixture Models

Using sender-specific and topic-specific structure models, we predict sender-topic-specific mixture models by interpolation:

$$P_{i,j}(l) = \alpha P_i^s(l) + (1-\alpha)P_j^t(l), \qquad (1)$$

where $P_{i,j}(l)$ is the estimated probability that the structural label l occurs from the sender i and for the topic j, $P_i^s(l)$ is the probability of the structural label l from the sender i (regardless of topics), $P_j^t(l)$ is the probability of the structural label l related to the topic j (regardless of senders), and α is the interpolation weight, balancing between sender style and topic focus. Figure 3 illustrates the mixture models combined by sender-specific and topic-specific models.

4.3 Generating Email Structure

We generate structural label sequences randomly according to the distribution from sender-topic-specific models. To generate the structural label sequences from the sender i and related to the topic j, the probability of the structural label l_k using n-gram language model is

$$P_{i,j}(l_k) = P_{i,j}(l_k \mid l_{k-1}, l_{k-2}, ..., l_{k-(n-1)}). \quad (2)$$

Since we use smoothed trigrams, we may generate unseen trigrams based on back-off methods, resulting in some undesirable randomness. We therefore exclude unreasonable emails that don't follow two simple rules.

1. The structural label "*greeting*" only occurs at the beginning of the email.
2. The structural label "*signature*" only occurs at the end of the email.

5 Surface Realization

Our surface realizer has four elements: building language models, generating text content, scoring email candidates, and filling slots.

5.1 Building Content Language Models

After replacing the tokens with slots, for each structural label, we train an unsmoothed n-gram language model using all sentences with that structural label. We make a simplifying assumption that the usage of within-sentence language can be treated as independent across senders; generating the text content only considers the structural labels. We use 5-gram to balance variability in generated sentences while minimizing nonsense sentences.

Given a structural label, we use the content language model probability directly to predict the next word. The most likely sentence is $W^* = \arg\max P(W \mid l)$, where W is a word sequence and l is a structural label. However, in order to introduce more variation, we do not look for the most likely sentence but generate each word randomly according to the distribution similar to Section 4.3 and illustrated below.

5.2 Generating Text Content

The input to surface realization is the generated structural label sequence. We use the corresponding content language model trained for the given structural label to generate word sequences randomly according to the distribution from the language model. The probability of a word w_i using the n-gram language model is

$$P(w_i) = P(w_i \mid w_{i-1}, w_{i-2}, ..., w_{i-(n-1)}, l), \quad (3)$$

where l is the input structural label. Since we build separate models for different structural labels, (3) can be written as

$$P(w_i) = P(w_i \mid w_{i-1}, w_{i-2}, ..., w_{i-(n-1)}) \quad (4)$$

using the model for l.

Using unsmoothed 5-grams will not generate any unseen 5-grams (or smaller n-grams at the beginning and end of a sentence). This precludes generation of nonsense sentences within the 5-word window. Given a generated structural label sequence, we can generate multiple sentences to create a synthesized email.

5.3 Scoring Email Candidates

The input to NLG contains the required information that needs to be in the output email, as described in Section 3.2. For each synthesized email, we penalize it if the email 1) contains slots for which there is no provided valid value, or 2) does not have the required slots.

The content generation engine stochastically generates an email candidate and scores it. If the email has a zero penalty it is passed on.

5.4 Filling Slots

The last step is to fill slots with the appropriate values. For example, the sentence "Tomorrow's [meeting] is at [location]." could become "Tomorrow's speech seminar is at Gates building."

6 Experiments

6.1 Setup

The corpus used for our experiments is the Enron Email Dataset[2], which contains a total of about 0.5M messages. We selected the data related to daily business for our use, including data from about 150 users. We randomly picked 3 senders, ones who wrote many emails, and defined additional 3 topic classes (meeting, discussion, issue) as topic-specific entities for the task. Each sender-specific model (across topics) or topic-specific model (across senders) is trained on 30 emails.

6.2 Evaluation of Sender Style Modeling

To evaluate the performance of sender style, 7 subjects were given 5 real emails from each sender and then 9 synthesized emails. They were asked to rate each synthesized email for each sender on a scale of 1 (highly confident that the email is not from the sender) to 5 (highly confident that the email is from that sender).

With $\alpha = 0.75$ in (1) for predicting mixture models (higher weight for sender-specific model), average normalized scores the corresponding senders receives account for 45%; this is above chance (which would be 33%). This suggests that sender style can be noticed by subjects, although the effect is weak, and we are in the process of designing a larger evaluation. In a follow-up questionnaire, subjects indicated that their ratings were based on greeting usage, politeness, the length of email and other characteristics.

6.3 Evaluation of Surface Realization

We conduct a comparative evaluation of two different generation algorithms, template-based generation and stochastic generation, on the same email structures. The average number of sentences in synthesized emails is 3.8, because our data is about daily business and has relatively short emails. Given a structural label, template-based generation consisted of randomly selecting an intact whole sentence with the target structural label. This could be termed sentence-level NLG, while stochastic generation is word-level NLG.

We presented 30 pairs of (sentence-, word-) synthesized emails, and 7 subjects were asked to compare the overall coherence of an email, its sentence fluency and naturalness; then select their preference. Table 1 shows subjects' preference according to the rating criteria. The word-based stochastic generation outperforms or performs as well as the template-based algorithm for all criteria, where a t-test on an email as a random variable shows no significant improvement but p-value is close to 0.05 ($p = 0.051$). Subjects indicated that emails from word-based stochastic generation are more natural; word-level generation is less likely to produce an unusual sentences from the real data; word-level generation produces more conventional sentences. Some subjects noted that neither email seemed human-written, perhaps an artifact of our experimental design. Nevertheless, we believe that this stochastic approach would require less effort compared to most rule-based or template-based systems in terms of knowledge engineering.

	Template	Stochastic	No Diff.
Coherence	36.19	**38.57**	25.24
Fluency	28.10	**40.48**	31.43
Naturalness	35.71	**45.71**	18.57
Preference	36.67	**42.86**	20.48
Overall	34.17	**41.90**	23.93

Table 1: Generation algorithm comparison (%).

7 Conclusion

This paper presents a two-stage stochastic NLG for synthesizing emails: first a structure is generated, and then text is generated for each structure element, where sender style and topic structure can be modeled. Subjects appear to notice sender style and can also tell the difference between templates using original sentences and stochastically generated sentences. We believe that this technique can be used to create realistic emails and that email generation could be carried out using mixtures containing additional models based on other characteristics. The current study shows that email can be synthesized using a small corpus of labeled data; however these models could be used to bootstrap the labeling of a larger corpus which in turn could be used to create more robust models.

Acknowledgments

The authors wish to thank Brian Lindauer and Kurt Wallnau from the Software Engineering Institute of Carnegie Mellon University for their guidance, advice, and help.

[2] https://www.cs.cmu.edu/~enron/

References

Regina Barzilay and Mirella Lapata. 2008. Modeling local coherence: An entity-based approach. *Computational Linguistics*, 34(1):1–34.

Regina Barzilay and Lillian Lee. 2004. Catching the drift: Probabilistic content models, with applications to generation and summarization. In *HLT-NAACL*, pages 113–120.

Anja Belz. 2005. Corpus-driven generation of weather forecasts. In *Proc. 3rd Corpus Linguistics Conference*.

Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proc. of ACL*, pages 363–370.

Alastair J Gill, Carsten Brockmann, and Jon Oberlander. 2012. Perceptions of alignment and personality in generated dialogue. In *Proceedings of the Seventh International Natural Language Generation Conference*, pages 40–48. Association for Computational Linguistics.

Irving J Good. 1953. The population frequencies of species and the estimation of population parameters. *Biometrika*, 40(3-4):237–264.

Shlomo Hershkop, Salvatore J Stolfo, Angelos D Keromytis, and Hugh Thompson. 2011. Anomaly detection at multiple scales (ADAMS).

Andrew Lampert, Robert Dale, and Cécile Paris. 2009. Segmenting email message text into zones. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, volume 2, pages 919–928. Association for Computational Linguistics.

Alice H Oh and Alexander I Rudnicky. 2002. Stochastic natural language generation for spoken dialog systems. *Computer Speech & Language*, 16(3):387–407.

Christina Sauper and Regina Barzilay. 2009. Automatically generating wikipedia articles: A structure-aware approach. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1-Volume 1*, pages 208–216. Association for Computational Linguistics.