Two-Stage Stochastic Email Synthesizer

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 presents the design and implementation details of an email synthesizer using two-stage stochastic natural language generation, where the first stage structures the emails according to sender style and topic structure, and the second stage synthesizes text content based on the particulars of an email structure element and the goals of a given communication for surface realization. The synthesized emails reflect sender style and the intent of communication, which can be further used as synthetic evidence for developing other applications.

1 Introduction

This paper focuses on synthesizing emails 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 emails are needed for other applications. For instance, our email synthesizer 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).

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. 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. Specifically we investigate whether stochastic techniques can be used to acceptably model longer texts and individual sender characteristics in the email domain, both of which may require higher cohesion to be acceptable (Chen and Rudnicky, 2014).

Our proposed system involves two-stage stochastic generation, shown in Figure 1, in which the first stage models email structures according to sender style and topic structure (high-level generation), and the second stage synthesizes text content based on the particulars of a given communication (surface-level generation).

2 The Proposed System

The whole architecture of the proposed system is shown in left part of Figure 1, which is composed of preprocessing, first-stage generation for email organization, and second-stage generation for surface realization.

In preprocessing, we perform sentence segmentation for each email, and then manually annotate each sentence with a structure element, which is used to create a structural label sequence for each email and then to model sender style and topic structure for email organization (1st stage in the figure). The defined structural labels include *greeting*, *inform*, *request*, *suggestion*, *question*, *answer*, *regard*, *acknowledgement*, *sorry*, and *signature*. We also annotate content slots, including general classes automatically created by named entity recognition (NER) (Finkel et al., 2005) and hand-crafted topic classes, to model text content for surface realization (2nd stage in the figure). The content slots include *person*, *organization*, *location*, *time*, *money*, *percent*, and *date* (general classes), and *meeting*, *issue*, and *discussion* (topic classes).

2.1 Modeling Sender Style and Topic Structure for Email Organization

In the first stage, given the sender and the focused topic from the input, we generate the email structures by predicted sender-topic-specific mixture models, where the detailed is illustrated as be-

Proceedings of the 8th International Natural Language Generation Conference, pages 99–102,
Philadelphia, Pennsylvania, 19-21 June 2014.

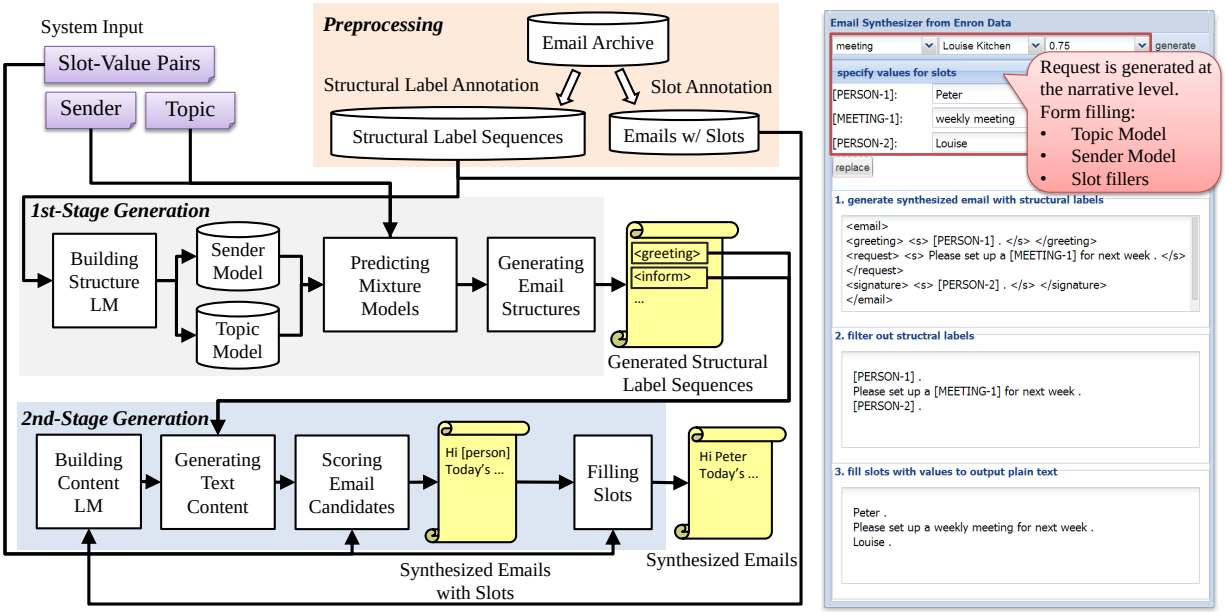

Figure 1: The system architecture (left) and the demo synthesizer (right).

low.

2.1.1 Building Structure Language Models

Based on the annotation of structural labels, each email can be transformed into a structural label sequence. Then we train a sender-specific structure model using the emails from each sender and a topic-specific model using the emails related to each topic. Here the structure models are trigram models with Good-Turing smoothing (Good, 1953).

2.1.2 Predicting Mixture Models

With sender-specific and topic-specific structure models, we predict the sender-topic-specific mixture models by interpolating the probabilities of two models.

2.1.3 Generating Email Structures

We generate structural label sequences randomly according to the distribution from sender-topic-specific models. Smoothed trigram models may generate any unseen trigrams based on back-off methods, resulting in more randomness. In addition, we 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.

2.2 Surface Realization

In the second stage, our surface realizer consists of four aspects: building content language models, generating text content, scoring email candidates, and filling slots.

2.2.1 Building Content Language Models

After replacing the tokens with the slots, for each structural label, we train an unsmoothed 5-gram language model using all sentences belonging to the structural label. Here we assume that the usage of within-sentence language is independent across senders and topics, so generating the text content only considers the structural labels. Unsmoothed 5-gram language models introduce some variability in the output sentences while preventing nonsense sentences.

2.2.2 Generating Text Content

The input to surface realization is the generated structural label sequences. We use the corresponding content language model for the given structural label to generate word sequences randomly according to distribution from the language model.

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

2.3 Scoring Email Candidates

The input to the system contains the required information that should be included in the synthesized result. 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 a candidate email, scores it, and outputs it when the synthesized email with a zero penalty score.

2.4 Filling Slots

The last step is to fill slots with the appropriate values. For example, the sentence "Tomorrow's [meeting] is at [location]." becomes "Tomorrow's speech seminar is at Gates building." The right part of Figure 1 shows the process of the demo system,where based on a specific topic, a sender, and an interpolation weight, the system synthesizes an email with structural labels first and then fills slots with given slot fillers.

3 Experiments

We conduct a preliminary experiment to evaluate the proposed system. The corpus used for our experiments is the Enron Email Dataset[1], which contains a total of about 0.5M messages. We selected the data related to daily business for our use. This includes data from about 150 users, and we randomly picked 3 senders, ones who wrote many emails, and define 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.

3.1 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 between 1 to 5.

With higher weight for sender-specific model when predicting mixture models, average normalized scores the corresponding senders receives account for 45%, which is above chance (33%). This suggests that sender style can be noticed by subjects. In a follow-up questionnaire, subjects indicated that their ratings were based on greeting usage, politeness, the length of email and other characteristics.

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

3.2 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. 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. The experiments showed that word-based stochastic generation outperforms or performs as well as the template-based algorithm for all criteria (coherence, fluency, naturalness, and preference). 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.

In the future, we plan to develop an automatic email structural label annotator in order to build better language models (structure language models and content language models) by increasing training data, and then improve the naturalness of synthesized emails.

4 Conclusion

This paper illustrates a design and implementation of an email synthesizer with two-stage stochastic NLG: first a structure is generated, and then text is generated for each structure element. Here sender style and topic structure can be modeled. We believe that this system can be applied to create realistic emails and could be carried out using mixtures containing additional models based on other characteristics. The proposed system shows that emails can be synthesized using a small corpus of labeled data, and the performance seems acceptable; 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.

References

Yun-Nung Chen and Alexander I. Rudnicky. 2014. Two-stage stochastic natural language generation for email synthesis by modeling sender style and topic structure. In *Proceedings of the 8th International Natural Language Generation Conference*.

Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 363–370. 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).

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