Harnessing Pre-Trained Neural Networks with Rules for Formality Style Transfer

Yunli Wang[†], **Yu Wu**[◇], **Lili Mou**[‡], **Zhoujun Li**[†]*, **Wenhan Chao**[†]
[†]State Key Lab of Software Development Environment, Beihang University, Beijing, China
[◇]Microsoft Research, Beijing, China
[‡]Dept. of Computing Science, University of Alberta, Edmonton, Canada
{wangyunli,lizj,chaowenhan}@buaa.edu.cn
Wu.Yu@microsoft.com doublepower.mou@gmail.com

Abstract

Formality text style transfer plays an important role in various NLP applications, such as non-native speaker assistants and child education. Early studies normalize informal sentences with rules, before statistical and neural models become a prevailing method in the field. While a rule-based system is still a common preprocessing step for formality style transfer in the neural era, it could introduce noise if we use the rules in a naïve way such as data preprocessing. To mitigate this problem, we study how to harness rules into a state-of-the-art neural network that is typically pretrained on massive corpora. We propose three fine-tuning methods in this paper and achieve a new state-of-the-art on benchmark datasets.

1 Introduction

Text formality research is essential for a wide range of NLP applications, such as non-native speaker assistants and child education. Due to the progress of deep learning techniques, researchers make a step from formality understanding to formality-aware text generation. Recently, Rao and Tetreault (2018) published a dataset, the Grammarly's Yahoo Answers Formality Corpus (GYAFC), serving as a benchmark testbed for formality style transfer, which aims to generate a formal sentence given an informal one, while keeping its semantic meaning.

Since the GYAFC dataset is small, existing studies have realized the importance of rules as a preprocessing step of informal text, typically handling capitalization (e.g., "ARE YOU KIDDING ME?"), character repetition (e.g., "noooo"), slang words (e.g., "wanna"), etc. While rule-based preprocessing could largely simplify the formality style transfer task, we observe that it also introduces noise, with an example shown in Table 1. Given a sentence with all capital letters, a common rule-based method would lower all characters except the first one. Some entities, such as R & B, are changed to lower case incorrectly, especially if not recognized as a proper noun.

Original informal sentence:
I LOVE HIP-HOP , RAP , ROCK & POP BUT MY FAV MUSIC IS R & B
Output of a rule-based system:
I love hip-hop , rap , rock and pop but my fav music is r and b

Table 1: Example of informal sentence and the output of a rule-based system.

Another intuition of ours is that, due to the small size of the parallel corpus, it would be beneficial to leverage a large neural network, which is pretrained on a massive corpus and has learned general knowledge of language. Then, we could fine-tune it in the formality style transfer task.

To this end, we study in this paper how to effectively incorporate pretrained networks—particularly, the powerful GPT-2 model (Radford et al., 2019)—with simple rules for formality style transfer. We analyze three ways of harnessing rules in GPT-2: 1) We feed the concatenation of the original informal sentence and the preprocessed one to the encoder; 2) We ensemble two models at the inference stage: one takes the original informal text as input, while the other takes the rule-preprocessed text as input; and 3) We employ two encoders to encode original informal text and the rule-preprocessed text seperately, and then develop a hierarchical attention mechanism in both word- and sentence-levels to aggregate information. Our work differs from previous work, which only feeds preprocessed inputs to the encoder. Rather, we are able to preserve more information of the original sentence, and the rule-based system is harnessed in a learnable way.

* Corresponding author. Our code and output are available at: https://github.com/jimth001/formality_emnlp19.git

Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, pages 3573–3578, Hong Kong, China, November 3–7, 2019.

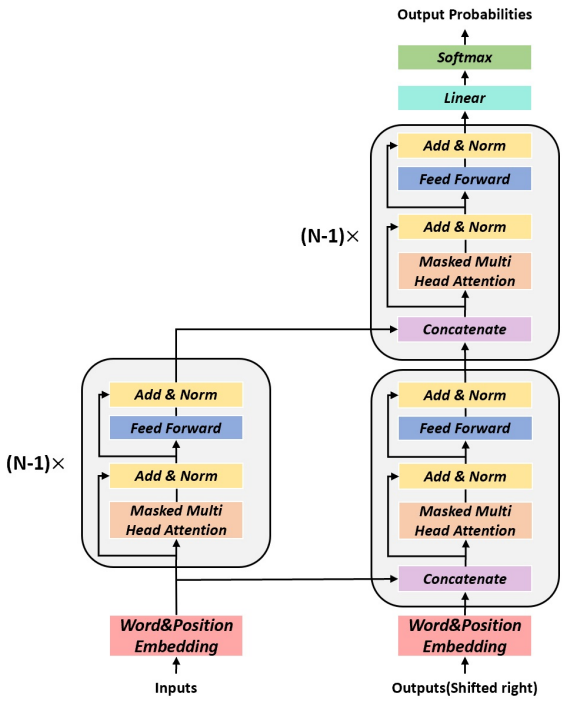

Figure 1: The architecture of our model. Both the encoder and decoder use masked multi-head attention blocks, but they do not share parameters. For the ith layer of the decoder (denoted as l_i^{dec}), it takes the concatenation of l_{i-1}^{enc} and l_{i-1}^{dec}'s outputs as input. N is the number of blocks in the decoder.

Experimental results show that our method outperforms direct fine-tuning of GPT-2 by 2 BLEU scores, and previous published results by 1.8–2.8 scores in different domains of the GYAFC dataset.

2 Related Work

In the past few years, style-transfer generation has attracted increasing attention in NLP research. Early work transfers between modern English and the Shakespeare style with a phrase-based machine translation system (Xu et al., 2012). Recently, style transfer is more recognized as a controllable text generation problem (Hu et al., 2017), where the style may be designated as sentiment (Fu et al., 2018), tense (Hu et al., 2017), or even general syntax (Bao et al., 2019; Chen et al., 2019). In the above approaches, the training sentences are labeled with style information, but no parallel data are given. Xu et al. (2019a) take one step further and capture the most salient style by detecting global variance in a purely unsupervised manner (i.e., style labels are unknown).

Formality style transfer is mostly driven by the GYAFC parallel corpus. Since a parallel corpus, albeit small, is available, formality style transfer usually takes a seq2seq-like approach (Rao and Tetreault, 2018; Niu et al., 2018a; Xu et al., 2019b). In particular, this paper focuses on harnessing pre-trained neural networks with rule-based systems.

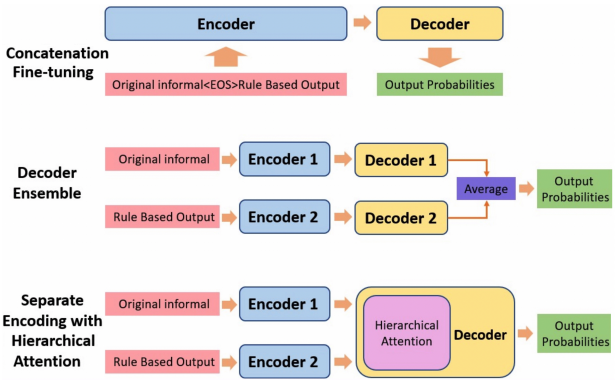

Figure 2: Three different methods to incorporate pre-trained models with rule-based systems.

3 Approach

We implement our encoder and decoder with GPT blocks, and initialize them with the pretrained GPT-2 parameters (Radford et al., 2019). The architecture of a decoder GPT block performs attention to the context words and previous words with the same multi-head attention layer, illustrated in Figure 1, which is slightly different with the classic Transformer (Vaswani et al., 2017). Formally, the output of the attention layer is[1]

$$\text{softmax}\left(\frac{\boldsymbol{Q}[\boldsymbol{K}_{\text{enc}};\boldsymbol{K}_{\text{dec}}]^T}{\sqrt{d_k}}\right)[\boldsymbol{V}_{\text{enc}};\boldsymbol{V}_{\text{dec}}] \quad (1)$$

where $\boldsymbol{Q}$, $\boldsymbol{K}$, and $\boldsymbol{V}$ are defined the same as the scaled dot-product attention in Transformer, and d_k is a scaling factor. $[;]$ is a concatenation operation; it enables to consider context words and previous decoding results in the same layer. Such architecture enables to adapt GPT-2, a Transformer-based pretrained language model, to a sequence-to-sequence model without re-initializing the parameters.

In the following, we describe several methods combining the GPT-based encoder-decoder model with preprocessing rules and (limited) parallel data.

Fine-Tuning with Prepocessed Text as Input. Given an informal sentence $\mathbf{x}_i$ as input, the most

[1] We use bold italic letters to denote a vector or matrix.

straightforward method, perhaps, is to first convert $\mathbf{x}_i$ to $\mathbf{x}'_i$ by rules, and then fine-tune the pre-trained GPT model with parallel data $\{(\mathbf{x}'_i, \mathbf{y}_i)\}_{i=0}^{M}$ (M is the number of samples). In this way, informal sentences can be normalized with rules before using a neural network. This simplifies the task and is standard in previous studies of formality style transfer (Rao and Tetreault, 2018).

However, the preprocessed sentence serves as a Markov blanket, i.e., the system is unaware of the original sentence, provided that the preprocessed one is given. This is in fact not desired, since the rule-based system could make mistakes and introduce noise (Table 1).

Fine-Tuning with Concatenation. To alleviate the above issue, we feed the encoder with both the original sentence $\mathbf{x}_i$ and the preprocessed one $\mathbf{x}'_i$. We concatenate the words of $\mathbf{x}_i$ and $\mathbf{x}'_i$ with a special token `EOS` in between, forming a long sequence $(x_{i,1}, \cdots, x_{i,s_i}, \texttt{EOS}, x'_{i,1}, \cdots, x'_{i,s'_i})$; after that, the concatenated sequence and the corresponding formal reference serve as a parallel text pair to fine-tune the GPT model. In this way, our model can make use of a rule-based system but also recognize its errors during the fine-tuning stage.

Decoder Ensemble. We investigate how the model performs if we train two GPTs with $\{\mathbf{x}'_i, \mathbf{y}_i\}_{i=0}^{M}$ and $\{\mathbf{x_i}, \mathbf{y_i}\}_{i=0}^{M}$ separately, but combine them by model ensemble in the decoding phase. We denote the generative probability of the jth word by $h(\mathbf{x}_i, \mathbf{y}_{i,<j})$ and $h'(\mathbf{x}'_i, \mathbf{y}_{i,<j})$. We apply "average voting" and the resulting predictive probability is $\frac{h'(\mathbf{x}'_i, \mathbf{y}_{i,<j}) + h(\mathbf{x}_i, \mathbf{y}_{i,<j})}{2}$.

Hierarchical Attention. In our final variant, we use two encoders to encode $\mathbf{x}_i$ and $\mathbf{x}'_i$ separately, but compute a hierarchical attention to aggregate information, given by

$$\alpha \cdot \text{softmax}\left(\frac{\boldsymbol{Q}[\boldsymbol{K}_{\text{enc1}}; \boldsymbol{K}_{\text{dec}}]^\top}{\sqrt{d_k}}\right)[\boldsymbol{V}_{\text{enc1}}; \boldsymbol{V}_{\text{dec}}] + \beta \cdot \text{softmax}\left(\frac{\boldsymbol{Q}[\boldsymbol{K}_{\text{enc2}}; \boldsymbol{K}_{\text{dec}}]^\top}{\sqrt{d_k}}\right)[\boldsymbol{V}_{\text{enc2}}; \boldsymbol{V}_{dec}] \quad (2)$$

where α and β are sentence attention weights for each decoding step, computed by

$$\alpha = \frac{\exp(\boldsymbol{h}_l^T \boldsymbol{W} \boldsymbol{z}_1)}{\exp(\boldsymbol{h}_l^T \boldsymbol{W} \boldsymbol{z}_1) + \exp(\boldsymbol{h}_l^T \boldsymbol{W} \boldsymbol{z}_2)}, \quad \beta = 1 - \alpha \quad (3)$$

Here, $\boldsymbol{h}_l$ is the hidden state of the lth step at the decoder, $\boldsymbol{W}$ is a learnable parameter, $\boldsymbol{z}_1$ and $\boldsymbol{z}_2$ represent the last hidden state of the two encoders, respectively. We propose this variant in hopes of combining the information of $\mathbf{x}_i$ and $\mathbf{x}'_i$ in the training stage and in a learnable way, compared with the decoder ensemble.

4 Experiments

4.1 Setup

We evaluate our methods on the benchmark dataset, the Grammarly's Yahoo Answers Formality Corpus (GYAFC, Rao and Tetreault, 2018). It consists of handcrafted informal-formal sentence pairs in two domains, namely, Entertainment & Music (E&M) and Family & Relationship (F&R). Table 2 shows the statistics of the training, development, and test sets. In the development and test sets of GYAFC, each sentence has four references, against which evaluation metrics are computed.

	Train	Dev	Test
Entertainment & Music	52,595	2,877	1,416
Family & Relationship	51,967	2,788	1,332

Table 2: Corpus statistics.

We implement our model with Tensorflow 1.12.0 and take the pretrained GPT-2 model (117M) released by OpenAI[2] to initialize our encoder and decoder. We use the Adam algorithm (Kingma and Ba, 2015) to train our model with a batch size 128. We set the learning rate to 0.001 and stop training if validation loss increases in two successive epochs.

4.2 Competing Methods

We compare our model with the following state-of-the-art methods in previous studies.

Rule-Based: We follow Rao and Tetreault (2018) and create a set of rules to convert informal texts to formal ones. Due to the lack of industrial engineering, our rule-based system achieves a performance slightly lower than (but similar to) Rao and Tetreault (2018).

NMT-Baseline: An RNN-based Seq2Seq model with the attention mechanism (Bahdanau et al., 2015) is trained to predict formal texts, given rule-preproccessed informal text.

PBMT-Combined: Similar to NMT, this baseline trains a traditional phrase-based machine translation (PBMT) system, also taking the preprocessed text as input. Then, self-training (Ueff-

[2] https://github.com/openai/gpt-2

ing, 2006) is applied with an unlabeled in-domain dataset for further improvement.

NMT-Combined: This method uses back-translation (Sennrich et al., 2016) with the PBMT-Combined system to synthesize a pseudo-parallel corpora. Then a Seq2Seq model is trained on the combination of the pseudo-parallel and parallel corpora.

Note that the above baselines are reported by Rao and Tetreault (2018).

Transformer-Combined: This setting in Xu et al. (2019b) is the same as NMT-Combined, except that it employs Transformer (Vaswani et al., 2017) as the encoder and decoder.

JTHTA: Xu et al. (2019b) propose a bi-directional framework that can transfer formality from formal to informal or from informal to formal with one single encoder-decoder component. They jointly optimize the model against various losses and call it Joint Training with Hybrid Textual Annotation (JTHTA).

Bi-directional-FT: Niu et al. (2018b) merge the training data of two domains and leverage data borrowed from machine translation to train their models with a multi-task learning schema, and also apply model ensembles. For fairness, we also combine the two domains when comparing with Niu et al. (2018b).

Additionally, we also evaluate our model variants. We first apply the GPT based on the original parallel corpus without using the rule-based system, denoted as GPT-Orig. Then, we feed the rule-preprocessed text as input, denoted as GPT-Rule. Other variants in Section 3 are denoted as GPT-CAT, GPT-Ensemble, and GPT-HA, respectively.

4.3 Evaluation Metrics

To evaluate different models, we apply multiple automatic metrics, mostly following Rao and Tetreault (2018).

Formality: Rao and Tetreault (2018) train a feature-based model to evaluate the formality of sentences, requiring an extra labeled corpus for training, which is unfortunately not publicly available. As a replacement, we train an LSTM-based classifier using the training data of GYAFC. It achieves 93% accuracy in the development and test sets, and thus is an acceptable approximation.

Meaning Preservation: We evaluate whether the meaning of the source sentence is preserved with a model trained on the Semantic Textual Similarity (STS) dataset. We adopt the BERT-Base[3] model (Devlin et al., 2019) and use STS for fine-tuning.

Overall: We evaluate the overall quality of formality-transferred sentences with BLEU (Papineni et al., 2002) and PINC (Chen and Dolan, 2011). BLEU evaluates the n-gram overlap, and PINC is an auxiliary metric indicating the dissimilarity between an output sentence and an input. A PINC score of 0 indicates that the input and output sentences are the same. According to Rao and Tetreault (2018), BLEU correlates with human annotation best.

4.4 Results

We show results of the E&M and F&R domains in Tables 3 and 4, respectively. We see that, by using the GPT-2 pretrained model alone (GPT-Orig), we achieve close results to previous state-of-the-art models. It outperforms NMT-Combine and JTHTA, even without fine-tuning on pseudo-parallel data. Our method also significantly outperforms the Transformer-Combined model (without pretraining). The results suggest that the small GYAFC corpus does not suffice to fully train the Transformer model. The GPT-2 model, pretrained with massive unlabeled corpora, is able to capture the generic knowledge of language and can be adapted to formality style transfer.

We then evaluate our different methods of incorporating the rule-based system into the pretrained GPT-2 model. We see that GPT-CAT yields the best results, which is probably because the concatenation enables two input sentences interact with each other through a single self-attention mechanism, while other methods encode each input sentences (original and rule-preprocessed) separately.

When combining both domains as in Niu et al. (2018b), we also have better performance than the previous work. This further shows the robustness of our model.

Regarding formality, our model achieves a reasonably high accuracy, although combining domains is slightly worse (since cross-domain training may bring noise that hurts the output formality). The rule-based model itself shows the best performance on content preserving, but it does not

[3] https://github.com/google-research/bert

	Formality	Meaning	BLEU	PINC
Original Informal	20.05	4.85	50.30	0
Formal Reference	79.61	3.78	100.00	66.93
In-domain data				
Rule-based†	48.69	**4.37**	60.35	28.26
NMT-baseline†	**77.38**	3.25	58.26	54.94
NMT-Combined†	73.81	3.88	67.55	43.45
PBMT-Combined†	66.94	4.00	66.87	43.27
Transformer-Combined	-	-	65.50	-
JTHTA	-	-	69.63	-
GPT-Orig	73.75	3.70	69.30	47.35
GPT-Rule	74.88	3.66	69.65	48.85
GPT-Ensemble	74.81	3.69	69.86	48.20
GPT-CAT	74.09	3.76	**71.39**	46.38
GPT-HA	74.37	3.67	69.03	47.82
Combined domains				
Bi-directional FT†	70.61	**3.98**	72.01	41.74
GPT-CAT	**71.45**	3.81	**72.70**	44.07

Table 3: Test performance on the E&M domain. PINC reflects the dissimilarity to the original informal sentence. Neither a too high nor a too slow score is desired. † indicates that we evaluate the output (if available) given by each paper with our metrics. Othewise, we quote the BLEU score from respective papers.

	Formality	Meaning	BLEU	PINC
Original Informal	21.31	4.76	51.66	0
Formal Reference	81.53	3.20	100.00	65.59
In domain data				
Rule-based†	57.50	**4.24**	66.36	27.75
NMT-baseline†	**79.31**	3.40	68.26	49.35
NMT-Combined†	76.75	3.77	73.78	41.76
PBMT-Combined†	77.45	3.82	72.40	44.02
Transformer-Combined	-	-	70.63	-
JTHTA	-	-	74.43	-
GPT-Orig	77.90	3.80	75.65	42.20
GPT-Rule	78.62	3.76	76.08	42.87
GPT-Ensemble	78.67	3.77	76.32	42.61
GPT-CAT	78.80	3.78	**77.26**	42.77
GPT-HA	77.31	3.79	76.31	41.86
Combined domains				
Bi-directional FT†	74.54	**3.97**	75.33	39.39
GPT-CAT	**76.84**	3.81	**76.87**	42.44

Table 4: Test performance on the F&R domain.

change the input much (a low PINC score).

In summary, our models significantly outperform previous work in formality style transfer and achieve a state-of-the-art performance on the two domains of GYAFC, which credits to both the pretrained model and our fine-tuning methods in consideration of a rule-based system.

5 Conclusion

In this work, we study how to incorporate a pretrained neural network with a rule-based system for formality style transfer. We find that building a pretrained GPT-2 upon the concatenation of the original informal text and the rule-preprocessed text achieves the highest performance on benchmark datasets.

Acknowledgments

The authors thank the anonymous reviewers for insightful comments. This work is supported in part by the National Natural Science Foundation of China (Grand Nos. U1636211, 61672081, and 61370126). Lili Mou is an Amii Fellow; he also thanks AltaML for support.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of the International Conference on Learning Representations*.

Yu Bao, Hao Zhou, Shujian Huang, Lei Li, Lili Mou, Olga Vechtomova, Xin-yu Dai, and Jiajun Chen. 2019. Generating sentences from disentangled syntactic and semantic spaces. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6008–6019.

David Chen and William B. Dolan. 2011. Collecting highly parallel data for paraphrase evaluation. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 190–200.

Mingda Chen, Qingming Tang, Sam Wiseman, and Kevin Gimpel. 2019. Controllable paraphrase generation with a syntactic exemplar. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5972–5984.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4171–4186.

Zhenxin Fu, Xiaoye Tan, Nanyun Peng, Dongyan Zhao, and Rui Yan. 2018. Style transfer in text: Exploration and evaluation. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, pages 663–670.

Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P. Xing. 2017. Toward controlled generation of text. In *Proceedings of the 34th International Conference on Machine Learning*, pages 1587–1596.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations*.

Xing Niu, Sudha Rao, and Marine Carpuat. 2018a. Multi-task neural models for translating between

styles within and across languages. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1008–1021.

Xing Niu, Sudha Rao, and Marine Carpuat. 2018b. Multi-task neural models for translating between styles within and across languages. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1008–1021.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*.

Sudha Rao and Joel R. Tetreault. 2018. Dear Sir or Madam, may I introduce the GYAFC dataset: Corpus, benchmarks and metrics for formality style transfer. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 129–140.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Edinburgh neural machine translation systems for WMT 16. In *Proceedings of the 1st Conference on Machine Translation (WMT)*, pages 371–376.

Nicola Ueffing. 2006. Self-training for machine translation. In *NIPS Workshop on Machine Learning for Multilingual Information Access*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 6000–6010.

Peng Xu, Yanshuai Cao, and Jackie Chi Kit Cheung. 2019a. Unsupervised controllable text generation with global variation discovery and disentanglement. *arXiv preprint arXiv:1905.11975*.

Ruochen Xu, Tao Ge, and Furu Wei. 2019b. Formality style transfer with hybrid textual annotations. *arXiv preprint arXiv:1903.06353*.

Wei Xu, Alan Ritter, Bill Dolan, Ralph Grishman, and Colin Cherry. 2012. Paraphrasing for style. In *Proceedings of the 24th International Conference on Computational Linguistics*, pages 2899–2914.