Unsupervised Controllable Text Formalization

摘要

我们提出了一种可控的自然语言转换的全新框架。我们意识到并行语言语料库的需求对于可控生成任务实际上是非可持续的，因此引入了一种无监督的训练方案。该框架的关键是一种深层神经编码器-解码器，该编码器通过辅助模块(称为计分器)来增强文本转换知识。这些计分器基于现成的语言处理工具实现，根据编码器-解码器的动作决定其学习的 Scheme。我们将这种框架应用于通过提高可读性等级正式化输入文本的文本转换任务；所需的正式化程度可以由用户在运行时控制。在公共数据集上进行的实验证明了本模型在以下方面的功效：(a)将给定文本转换为更正式的风格,(b)根据指定的输入控制改变输出文本中形式性的量。我们的代码和数据集已公开发布供学术使用

1. 引言

自动文本风格转换是*文本到文本*的自然语言生成(NLG)研究的关键目标之一，并且大多数用于此类任务的现有系统都是有监督的(例如，*Seq2Seq* 神经模型的各种版本(Sutskever, Vinyals 和 Le, 2014; Bahdanau, Cho 和 Bengio, 2014)或统计机器翻译模型(Koehn, 2009)或基于模板/规则的(Gatt 和 Reiter 2009)。在受有监督的 NLG 中，需要大规模的并行语料库进行训练，在扩展到各种实际用例时这是其主要障碍(impediment)。例如，仅在商业对话系统的情景下，就有多种需要对系统回答(可能来自数据库(Jain 等人, 2018))的*语气*(礼貌，兴奋等)，或*正式程度*(休闲，正式等，这基于用户的性格)，或*复杂性*(使用简单化的表述或使用如法律或医学领域的特定领域术语)进行转换的情景。随着此类需求和用例的不断增长，为每种此类文本转换任务都构建大规模平行语料实际上成为了非可持续的任务。

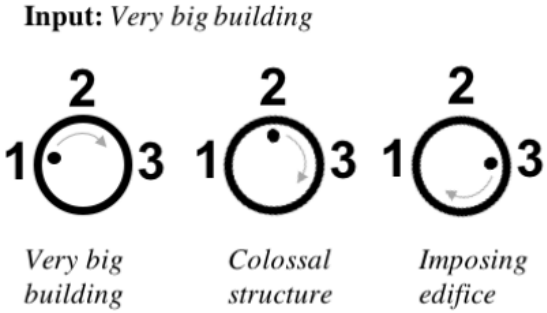

Figure 1: Controllable formalization showing an anecdotal snippet and its expected variations w.r.t. input controls

从科学的角度来看，使用多个平行语料库对所有这些任务进行监督处理，旨在(在保留语义的情况下)学习语言转换的同时学习风格转换。我们对此有 3 个主要的发现：(i)由于对于语言转换必须的保留原文的语义，而仅需改变文本的特征属性或风格，因此应该可以将这两件事解耦,(ii)相较于为带有输出示例的每个输入文本*指定*所需的转换，在输出端(使用易于

理解的 NLP 技术)分开*处理(verify)*这两方面要经济许多；并且(iii)应该可以按输出的需要来*控制*意图属性的等级或极性(如可读性，礼貌程度等)。这些发现结果促使我们寻求一种无监督的方法来全方位解决文本转换任务，并为此构建适合的 NLG 框架。

我们提出的框架仅需要用于初始化的无标记文本和一组*现成的*语言处理模块。我们在*输入文本正式化*的任务下测试了本框架，该框架在保持输出文本含义的同时提高了输出文本的可读性(请参见图 1)。所需的正式程度可以由用户在运行时确定，并作为系统的控制输入。选择此任务是因为它与许多 NLG 应用程序相关，例如正式会话生成，电子邮件回复撰写或巡查和监管领域中合规的摘要文档生成等。不仅如此，这种独立的系统可有像计算机辅助翻译(CAT)系统现在为翻译家提供帮助一样协助专业作家。这为低成本和高效的文本内容创建解决方案铺平了道路。

本框架基于一个编码器-解码器模块(Bahdanau, Cho 和 Bengio 2014)，它是用未标记的文本预先训练的。**解码器还额外将用户指定的控制参数作为输入**。此外，通过辅助模块(称为计分器)获得所需文本转换的知识，辅助模块基于编码器-解码器的动作来决定编码器-解码器的学习 Scheme。这些计分器基于易得的自然语言处理(NLP)工具，这些工具可以产生表示以下内容的评分:(a)生成文本的正式程度,(b)生成文本的流畅程度，最重要的是(c)生成的文本是否带有与输入文本相似的语义。这个框架在多次迭代中进行训练，每次迭代包括两个阶段(I)**探索**和(ii)**开发**。在探索阶段，解码器随机对给定输入的候选文本进行采样，并在计分器的帮助下，为可控生成任务自动生成训练数据。在开发阶段，编码器-解码器用刚才生成的数据重新训练。

关于实验，我们准备了一份低可读性的无标记通俗文本，并通过我们的 NLP 工具确保了可读性、适当性(adequacy)和流利性。通过这种配置，我们发现到(a)系统生成的转换文本比输入更正式,(b)它们的正式程度符合用户给出的输入控制。系统的效率通过人类判断和各种 NLG 评估指标来定性和定量地评估和证明。我们还展示了该系统对另一个相关的文本深化(text complexification)任务(反向简化)的有效性，其源代码和数据公开。

2. 相关工作

由于以下事实，无监督的 NLG 总是更具挑战性:(1)输出空间更加复杂和结构化，使得无监督学习更加困难,(2)在没有参考输出文本的情况下评估 NLG 系统的指标难以捉摸。最近，Artetxe 等人(2017)和 Lamble、Denoyer 和 Ranzato(2017)提出了基于使用词汇归纳(lexicon induction)技术自编码器的无监督语言翻译架构。该方法主要关注跨语言转换，且需要来自不同语言的多个无标记语料。它不能简单地扩展到我们的环境，以在单一语言中实现可控的文本转换这一目标，而且语言翻译任务中也没有控制标记(notion)。Hu 等人(2017)的一份重要工作讨论了可控生成；该系统接收情绪、时态等控制参数，并生成符合条件的随机句子。然而,这个系统与本系统的不同之处，在于它不转换给定的输入文本。

与我们最相关的工作是由(Mueller, Gif-ford, 和 Jaakkola, 2017)联合训练的 VAE 及一个校正输入的效果预测模块，这个模块使文本输出更好的效果。我们采用了他们的系统配置进行了正式文本风格转换实验来比较系统性能。结果发现这个模型不像本模型那样接受外部控制参数。

其他一些相关的工作是关于基于情感和属性的无监督风格转换(Ficler 和 Goldberg, 2017; Shen 等人，2017; Shen 等人，2017)，通过使用翻译语料的反向翻译进行半监督转换(Prabhumoye 等人, 2018)；使用语言特征进行正式-非正式文本分类(Sheika 和 Inkpen, 2012)；礼貌程度分析(Danescu-Niculescu-Mizil 等人，2013)；和使用编码器-解码器模型进行礼貌对话生成(, 2018)，但这些模型无法进行可控的文本转换。

其他相关生成框架，如(Sheika 和 Inkpen, 2011)的正式文本生成和(Wubben, VanDen Bosch, 和 Krahmer, 2010; Prakash 等人, 2016)的释义(paraphrase)生成，都很类似，他们不是基于模板就是有监督，且均不可控的。(Li 等人, 2017)和(Yu 等人, 2017)的语言生成系统引入了基于 NLP 的评分器，但(Hu 等人, 2017)也指出其在训练过程中存在收敛问题。据我们所知，我们的工作是第一个实现*无监督可控文本转换*任务的方法

3. 系统概览

本框架被设计成接受文本(或句子)和一组控制参数。现在我们只考虑一个输入控制，比如正式化程度。对于这样的任务，神经编码器-解码器是一个自然的选择。在我们的设置中，将控制水平值与输入文本分开输入编码器-解码器。为了在无监督的情况下训练该模块，需要使用额外的组件。图 2 描述了系统和学习 scheme 的概览。

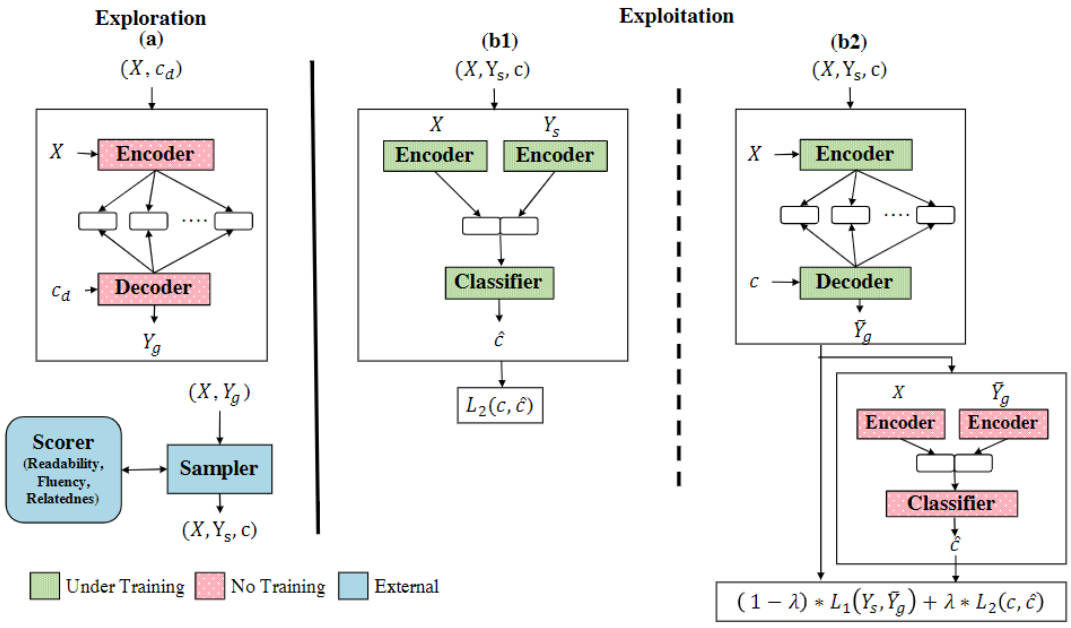

Figure 2: System architecture and phases of training

3.1 编码器和可控解码器

本框架基于编码器-解码器架构(Bahdanau, Cho 和 Bengio, 2014)。这种方法的工作原理是众所周知的，因此在本文中不再赘述。我们与传统架构的一个重要的不同是在解码器中增加了一个额外的输入——控制输入。控制输入被传入d维的嵌入层。

给定一个包含N个单词的输入语句X和c维控制向量，编码器首先使用堆叠的 GRU 层产生一个隐藏表示H(Cho 等人，2014)。解码器在每个时间步长产生一个单词，方法是:(a)attending 构建上下文向量的隐式表示,(b)将带有控制向量的上下文向量和解码器的当前隐状态相连接，以及(c)通过非线性函数从结果输出产生条件概率分布。总而言之，对于M个单词的输出句子$Y = (y_1, y_2, \ldots, y_M)$，每个单词$y_i$的条件概率可计为:

$$P(y_i|\{y_1, y_2, \ldots, y_{i-1}\}, z_i, c) = g(Ey_{i-1}, c, s_i, z_i)$$

其中g为输出y_i概率分布的 softmax 函数; Ey_{i-1}是前一个生成词的嵌入; c是控制向量; z_i是通过 attending H得到的向量; s_i是解码器 GRU 在第i步的隐状态。

3.2 采样器和计分器

为了以无监督方法(即没有可用的真实标签Y)训练编码器-解码器框架，输出概率分布传入了从解码器中采样K个不同样本的采样器，而不是获取解码器的最大似然(most likely)输出。采样对于训练过程中的探索步骤是必需的(稍后在第 4.2 节中讨论)。给定一个输入文本$\boldsymbol{X}$，编码器产生输出$\boldsymbol{Y}_g$。采样器的作用是接收Y_g并通过施加微小的改变来产生K种$\boldsymbol{Y}_g$的版本。一旦有了K个样本，最大化加权分数(由计分器模块产生)的最佳样本($\boldsymbol{Y}_s$)将被留存以供进一步处理。

$$\boldsymbol{Y}_s = \underset{Y}{\operatorname{argmax}}\{G(\boldsymbol{X},\boldsymbol{Y})|\boldsymbol{Y} \in \{\boldsymbol{Y}_g, Sample_K(\boldsymbol{Y}_g)\}\}$$

其中$G(\cdot)$是计分器函数，$Sample_K(\cdot)$是产生K个输出的采样器函数。

计分器$G(\cdot)$由多个独立的评分模块组成，这些评分模块评估生成文本的各方面的指标。在我们的设置中，计分器会衡量:(a)生成的文本在多大程度上与输入语义相关,(b)生成的文本有多流畅,(c)就可读性等级而言，文本有多正式。这些方面的测量方法如下:

语义相关性：源文本($\boldsymbol{X}$)和生成/采样的文本($\boldsymbol{Y}$)之间的语义相关性(记为r_s)表示了在转换过程中语义保持得有多好。

$$r_s(\boldsymbol{X},\boldsymbol{Y}) = docsim(\boldsymbol{X},\boldsymbol{Y})$$

其中$docsim(\cdot)$可以是任何文档相似性的度量。为了简化我们使用基于嵌入的相似度，其为$\boldsymbol{X}$和$\boldsymbol{Y}$的平均嵌入向量之间的余弦相似度(这是 Spacy(spacy.io)文档相似度函数。文本之间的相似度是在删除停用词后统计的)。

流畅性：流畅性(记为r_f)即是目标文本($\boldsymbol{Y}$)的结构构建得如何，通过语言模型来衡量。

$$r_f(\boldsymbol{Y}) = p(y_1, y_2, \dots, y_M)$$

可以分解成条件概率项的乘积。为了估计条件概率，可以使用 N-gram(Brown 等人, 1992)或神经语言模型(Bengio 等人, 2003)。在我们的实验中，我们使用 KenLM(Hefield, 2011)和包含混合领域文本的 Europarl *单语言*英文语料库(Koehn, 2005)为之训练了一个带回退的 4-gram 模型。

可读性：可读性等级得分(记为r_d)表示理解给定文本必须升入的年级(grade level that must be acquired to understand a given text)。通常更高的分数表明更高的语言学复杂性。在几个可用的阅读能力等级评分指标中，我们选择了 Flesch-Kincaid 可读指数(Kincaid 等人, 1975)，这是一个简单而普遍的可读性指标。度量标准依赖于句子中的字数，以及每个单词的平均音节数；这个指标刻画了文本词法复杂性。

累积计分函数$G(\cdot)$是上面讨论的三个指标得分(得分已归一化到$[0,1]$)的线性组合:

$$G(X,Y) = \beta_s r_s(\boldsymbol{X},\boldsymbol{Y}) + \beta_f r_f(\boldsymbol{X},\boldsymbol{Y}) + \beta_d r_d(\boldsymbol{X},\boldsymbol{Y})$$

其中权重β_s、β_f和β_d均由经验决定。

3.3 采样策略

在生成阶段的第i步中，不是从$\boldsymbol{y}_i$中选择最可能的单词，而是随机选择一个它的同义词。这个词的同义词是通过查找同义词表(我们用 WordNet 英文语料构建了同义词表)，并以均匀概率[*1/同义词总数*]随机选取一个同义词，一个句子就是由这样选出的单词串联而成。因为同义词的可能组合数量巨大，所以当抽取了K个样本句后就停止这次采样的迭代。

请注意，我们的采样器是高度词法化的(lexicalized)，即它只产生词法变化。做出这种选择的背后原因有两个个:(a)大多数现有的评估可读取性和文档相似性的计分器均强调词法层面,(b)在语法和语义层面，转换文本的通用/数据无关转换器(paraphrasers)难以理解(elusive)。然而，采样器在外部起作用，随着技术水平的提高，更换更好的采样器就可以提升本系统性能。例如，为了获得更多样的样本，可以考虑使用在无标记文本上训练的 VAE(Sohn, Lee, Yan, 2015)。

3.4 确定控制参数

采样器和计分器为(X, Y_s)产生新的示例(example)输出。然而，为了训练编码器-解码器(见第 4.3 节的*开发阶段*)，系统需要以$(\boldsymbol{X}, \boldsymbol{Y}_s, c)$形式输入的数据。新生成的示例其控制水平值$(c)$由如下方式确定:

$$c = \begin{cases} 1, if\ c_r < \zeta_1 \\ 2, if\ \zeta_1 < c_r < \zeta_2 \\ 3, if\ c_r > \zeta_2 \end{cases}$$

其中$c_r = r_d(Y_s)/r_d(X)$。

在我们的设置中，我们只接受一个可读性的控制参数。对于需要更多参数的场景或其他任务，可以在同样的框架内定义它。

3.5 控制预测器

训练期间编码器-解码器将与一个分类模块，或称*控制预测器*耦合。控制预测器的作用是预测任意输入的输出对$(\boldsymbol{X}, \boldsymbol{Y})$的期望控制水平值。控制预测器在嵌入层的顶部构成一对编码器，它们分别从$\boldsymbol{X}$和$\boldsymbol{Y}$得到隐藏表示$\boldsymbol{h_x}$和$\boldsymbol{h_y}$。然后这两个隐藏表示被拼接起来得到$\boldsymbol{h} = [\boldsymbol{h_x}; \boldsymbol{h_y}]$，$\boldsymbol{h}$被传递给使用 Relu 的全连接神经网络以获得中间隐藏表示$\boldsymbol{h_f}$。为了获得概率向量，我们对$\boldsymbol{h_f}$进行线性变换并使用 softmax 对其进行归一化，如下所示:

$$\boldsymbol{h_f} = relu(\boldsymbol{W_h h})$$

$$\boldsymbol{h_c} = softmax(\boldsymbol{W_c h_f})$$

在训练期间，控制预测器接收采样器和计分器模块生成的形如$< \boldsymbol{X}, \boldsymbol{Y}_s, c >$的实例。它给出预测控制水平值$\hat{c}$，采用基于*交叉熵*的损失函数$\boldsymbol{\mathcal{L}_2}(\boldsymbol{c}, \hat{\boldsymbol{c}})$计算损失并反向传播回嵌入层。第 4.4 节的需求促使了本模块的诞生。

4. 训练目标与流程

训练分三个阶段进行:(1)*预训练*(2)*探索*和(3)*开发*。当阶段(1)执行完成后，阶段(2)和(3)将会在系统中多次迭代执行。图 2 给出了训练过程的概览。

4.1 预训练

编码器-解码器首先作为使用无标记数据的自编码器进行预训练，它学习从X中预测X'即$X' \sim X$。由于在该阶段没有控制水平值可输入，因此控制输入被忽略。预训练确保了编码器和解码器参数被更好地初始化。

4.2 探索(Exploration)阶段

标记数据不能直接用于训练，因此就用探索阶段来合成包含输入文本、输出文本和控制参数$(< \boldsymbol{X}, \boldsymbol{Y}, c >)$的实例。系统在此过程中仅预测输出$\boldsymbol{Y_g}$而不进行训练，并“探索”其他与$\boldsymbol{Y_g}$不同的样本。在采样器的帮助下，系统会生成$K$个样本输出，然后计分器将选择能最大化分数的样本$\boldsymbol{Y_s}$。如果其他样本都比默认输出$\boldsymbol{Y_g}$得分低，那么本轮探索迭代中就不对这一输入$X$进行数据增强。如果生成的输出与输入相同，丢弃之。用于生成样本$\boldsymbol{Y_s}$的可能控制水平值是根据下式计算的。

$$c = \begin{cases} 1, if\ c_r < \zeta_1 \\ 2, if\ \zeta_1 < c_r < \zeta_2 \\ 3, if\ c_r > \zeta_2 \end{cases}$$

由于对编码器-解码器是预训练过的，在第一次探索迭代中，模型预测值$\boldsymbol{Y_g}$与输入X是一摸一样的。由于计分器选择的采样句得分总是高于$\boldsymbol{Y_g}$(否则不会被选择)，第一次探索迭代确保输出端的生成数据不同于输入端，且比输入端的累计得分要高。

4.3 开发(Exploitation)阶段

在此阶段，编码器-解码器使用探索阶段生成的数据进行训练。该过程分如下两步进行:

训练控制预测器：控制预测器模块通过输入X, Y并输出预测值c来进行训练。它在标准分类任务设置下进行训练——标记数据的 batch 以多次迭代的方式输入并最小化损失。一旦训练完成，预测器就被插入到编码器-解码器网络中(在那里它的参数置为固定值)，以预测对生成的句的控制水平值。因为控制水平值表示了$\boldsymbol{Y}$关于$\boldsymbol{X}$的**相对**差异，所以$\boldsymbol{X}, \boldsymbol{Y}$都被输入预测器。

训练编码器-解码器：编码器-解码器框架使用源文本$\boldsymbol{X}$、控制水平值c和目标输出$\boldsymbol{Y_g}$进行训练。模型在每个数据实例上输出预测值$\boldsymbol{Y_s}$，然后控制预测器预测输出($\boldsymbol{Y_g}$)的控制类别。然而，得出$\boldsymbol{Y_g}$需要使用 *argmax* 操作找出条件概率分布(式 1)中最可能的词。这使得全局损失（式 7)不可微。为了避免这种情况，我们用如下方法逼近$\boldsymbol{Y_g}$:

$$\boldsymbol{Y_g} \approx \widehat{\boldsymbol{Y}}_{\boldsymbol{g}} = \{\tilde{y}_1, \tilde{y}_2, \dots, \tilde{y}_M\}$$

其中，$\tilde{y}_i \sim softmax(s_i/\tau)$，$s_i$是未归一化的解码器隐藏表示(logits)，$\tau > 0$是*温度(temperature)*参数。将温度置为趋近于零的数就可以将输出近似为一般得通过 argmax 才能得出的独热表示。不同于 argmax，这确保了可微性。

修正输出$\widehat{\boldsymbol{Y}}_{\boldsymbol{g}}$是控制预测器的输入，它对于预训练的$\boldsymbol{X}$和$\boldsymbol{Y_g}$预测适当的控制水平值$\hat{c}$。基于输出的预测值和控制水平值，可如下计算出损失:

$$\boldsymbol{\mathcal{L}}\left(\boldsymbol{Y_s}, \boldsymbol{Y_g}, \boldsymbol{c}, \hat{\boldsymbol{c}}\right) = \lambda \times \boldsymbol{\mathcal{L}_2}(\boldsymbol{c}, \hat{\boldsymbol{c}}) + (1-\lambda) \times \boldsymbol{\mathcal{L}_2}(\boldsymbol{Y_s}, \widehat{\boldsymbol{Y}}_{\boldsymbol{g}})$$

其中$\lambda \in [0, 1]$是超参数，函数$\boldsymbol{\mathcal{L}_2}(.)$和$\boldsymbol{\mathcal{L}_1}(.)$是基于交叉熵的损失函数，他们通常被用于 Seq2Seq 和分类任务中。

4.4 为何使用控制预测器

在我们的转换设置中，单词层面的熵有如下问题: (1)单词级的损失在生成的单词上均等地惩罚所有单词；然而在可控制的转换中，一些词(例如内容词)比其他词(例如停用词)起着更重要的作用，(2)仅仅使用交叉熵不足以描述生成的句子是如何与传入的控制水平值相关联的。为了缓解这些问题，我们使用了一个预测器，该预测器能得出生成句的控制水平值的预测值与参考值之间的额外损失。

5. 实验设置

本节描述所用数据集、所选架构和所选超参。我们的数据集包含简单且通俗的无标记自然语言文本。它由 14432 个从 Enron Email Corpus、Corpus of Late Modern English Prose、non-spam emails from Spam Dataset 和 essays for kids 中收集的大量通俗句子组成。数据分成被划分为训练集:验证集:测试集= 80%: 12%: 8%。数据集的词表大小为 95775 词。源代码和数据集可在 https://github.com/parajain/uctf 处获得

我们使用了两层基于双向 GRU 的编码器；编码器的前向和后向隐藏表示被拼接在一起。编码器和解码器共享 300 维的词嵌入向量。对于用了两层 GRU 的编码器，隐层维度为 250。解码器有两层，隐层维度为 500。式 3 中给出的与语言模型、句子相似性和可读性相关的累计计分器，其权重根据经验分别设置为 0.6、0.2 和 0.2，但式 3 允许不同的转换参数的组合，这些参数可以根据实际使用进行不同的调整。例如，低权重的语义相关性对“相关性”的限制会更小且可以生成具有更高可读性的文本，这适用于诸如文本创作这样的领域；另一方面，

对于法律等语义相关性更重要的领域，可将相应的权重调高。

在损失计算(等式 7)中，我们尝试了不同的参数$\lambda \in [0.1,0.3,0.5]$，最后决定使用 0.1。式 4 中的$\zeta_1$和$\zeta_2$分别置为 1.05 和 1.1；每个类别的阈值提高了 5 %以符合实际目的。温度参数τ置为 0.001；我们没有使用退火(annealing)。对于编码器-解码器和控制预测器，使用了学习率为 0.001 的 Adam 优化器。系统先进行一次预训练，随后进行*最多* 20 期的探索-开发迭代。对于每个输入，选择最大$K = 100$个样本。在 20 期迭代中如果发生早停现象则执行开发阶段。在每期结束后，模型参数均被保存，我们从中选择了获得平均分最高的那个模型，该得分在仅用于模型选择而准备的保留集上得出。在测试过程中，测试数据中的每个输入句子其控制水平值仅可能取 2 或 3。

测试场景：本系统当前被配置为接收一条输入语句和有**三个可能取值**的控制等级，控制水平值被预训练初始化为用户需要的正式化等级。控制等级取值为 1、2 和 3。如引言部分图 1 所示，输入控制等级值 1 会将保持输入句子原样，控制等级 2 将把句子转换成*相对*正式的文本(称为*中等正式*)，而控制等级 3 将把输入文本转换成比等级 2 更正式的文本(称为*高等正式*)。

5.1 解决的研究难题

我们的实验旨在解决以下研究难题:

- **RQ1**：我们的方法能够产生正式、流畅和恰当的输出吗？这种转变是可控的吗？输出的正式程度与给出的输入控制有多相合？
- **RQ2**：控制预测器对我们模型的整体性能有积极影响吗？
- **RQ3**：探索-开发迭代有用吗？为什么更简单的单轮采样+训练策略是不够的？
- **RQ4**：对于这些任务，我们的方法是否优于现有的有监督和无监督方法？

RQ1 的答案是根据 3.2 节计算的输出文本的流畅性、语义相关性和可读性。为了找到 **RQ2** 和 **RQ3** 的答案，我们准备了两个简单的不同版本系统：(a)CTRLNOPREDICTOR，它移除了控制预测器模块，以及(b)CTRLONESHOT，它的勘探和开发周期不会迭代地进行。相反该系统进行一期探索，就紧跟着进行一期开发。

对于 **RQ4**，因为现有的方法/系统没有提供像我们这样的控制参数，所以对比只能基于它们的转换能力进行。

与现有无监督方法的比较:最相关的无监督转换系统是米勒、吉福德和雅克科拉(2017)。该系统以一个序列作为输入，对其进行自动编码，并根据主持人决定的预期结果应用最小化。我们将可读性度量作为评分函数，并将系统作为基线。

与现有的无监督方法比较：最相关的无监督转换系统是 Mueller, Gifford, and Jaakkola (2017)的系统。该系统以一个序列作为输入，对其进行自动编码，并基于计分器决定的预期效果施加微小的变化。我们将可读性指标作为计分函数并将该系统作为基线系统。

与有监督系统的间接比较：对本系统来说，一个相似的任务是有带标注数据的反向文本简化任务。我们选择了 Nisioi 等人(2017)在神经文本简化工作中给出的数据集配置，该数据集最初来自 Hwang 等人(2015)。我们翻转了平行语料的输入端和输出，从而反转了训练目标。我们的系统仅使用数据的简化侧(simplified side of the data)进行训练，并采用前文所述的配置。此外，使用了 Marian 工具包(Junczys-Dowmunt 等人, 2018)按默认配置(70000 次迭代后收敛)在完整的简*单到复杂*平行语料上训练了神经 Seq2Seq 生成系统(Bahdanau, Cho 和 Bengio, 2014)。

6. 实验结果

在我们的数据集上，我们首先基于以下参数评估了输出: (a)基本参数，例如流畅性、适当程度(输入的语义相关性)和可读性等级，以及(b)输出质量是否符合输入的控制参数。使用这些指标可以对本系统的不同版本进行直接比较。

Mode	CTRL WITHPREDICTOR		CTRL NOPREDICTOR		CTRL ONESHOT		Mueller et al., 2017
Formalness Control	Mid	High	Mid	High	Mid	High	NONE
Readability	0.568	0.583	0.538	0.538	0.554	0.554	0.33
Relatedness	0.72	0.74	0.77	0.77	0.78	0.78	0.05
LM Score	0.34	0.34	0.32	0.32	0.30	0.30	0.16

Table 1: Average test-set scores (normalized between $[0-1]$)

从表 1 中可以看出, CtrlWithPredictor 的可读性分数总是比输入(0.54)的平均可读性好。这表明系统获得了能改进可读性的转换能力。CtrlWithPredictor 的提升甚至比另外两个版本更大，这说明了控制预测器和迭代训练方案的重要性。CtrlWithPredictor 的语言模型部分得分最高，且显著差异优于 CtrlNoPredictor。这表明加入来自控制预测器的辅助损失可以得到更高的流畅性。就语义相关性而言，我们发现大多数由 CtrlOneShot 生成的句子相较于输入文本几乎没有或根本没有经过更改，这导致语义相关度得分较高。此外，在大多数情况下，从这两个系统获得的输出对控制水平值(*高等正式度*和*中等正式度*)不敏感，因此两个控制水平下其平均得分持平。对于相同的输入文本，CtrlOneShot 同一输入的多个版本是一次性(one-shot)生成的。事后想来，在训练过程中使用这些数据可能会误导系统，让它学会忽略控制输入。最后，CtrlNoPredictor 几乎在预训练的控制值下无法产生直接损失，因此它与控制水平值的关系成为了黑箱。

System	BLEU	Relatedness (with input)	Readability
Mueller et al.	5.09	0.41	0.38
Seq2Seq (skyline)	38.37	0.17	0.71
Formalness-Mid (Ours)	21.81	0.58	0.52
Formalness-High (Ours)	21.14	0.57	0.74

Table 2: Comparison of BLEU (%), avg. Semantic Relatedness with input, and avg. normalized readability scores. **The avg. readability for input and reference sentences in test data are (0.41) and (0.50) respectively**

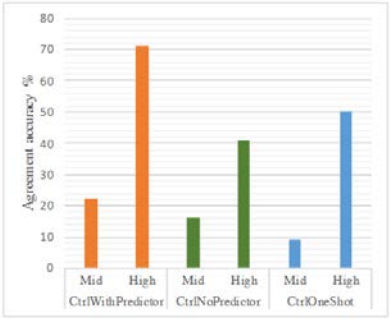

Figure 3: Accuracy (in %) showing the agreement between desired input control and control measured on the output text using Equation 4

图 3 展示了系统对输入控制的响应有多健壮。图中的准确率(agreement accuracy)指的是输出的控制的测量值 (由式 4 得出)与实际输入的控制值的匹配程度。对于所有三个系统，若控制水平值为*中等*，那么大多数生成的文本实际上要么与*默认*(即非常类似于输入)水平类似，要么与*高等水平*糅杂不清。这表明需要对式 4 来对用户定义的超参数进行精调。但请注意, CtrlWithPredictor 的准确率是最高的。

6.1 人类评估

我们进行了一项额外实验，3 名不了解该实验的语言学家从我们的测试集中得到了 30 个随机样本。每个数据实例都由输入语句和带有中等或高级控制水平值(为了去除偏差而进行了重排)的预测输出组成。对于每个实例，专家们都被要求根据他们认为的可读性标准对

输出句子进行评价。理想情况下，有中等控制输出文本的评价应低于带有高级控制输出文本的评价。人工评价给出的标签和基于输入控制输出的标签的平均一致性为 80.2%，这表明*高级*输出的可读性比的确比*中等*要高。

Mode	Input Sentence	*Formalness Control-Mid*	*Formalness Control-High*
WithPred	(1) 18 year old who abandoned her child in a hospital later got custody	(1) 18 year old who unpopulated her kid in a infirmary resultant got custody	(1) 18 year old who deserted her tyke in a infirmary resultant got detention
	(2) the first sync after upgrading will be slow	(2) the first synchronise afterward upgrading will be idle	(2) the first synchronise afterward upgrading will be laggard
NoPred	(1) 18 year old who abandoned her child in a hospital later got custody	(1) 18 class old who deserted her child in a infirmary accompanying got detention	(1) 18 class old who deserted her child in a infirmary accompanying got detention
	(2) the first sync after upgrading will be slow	(2) the introductory synchronise afterward upgrading will be goosy	(2) the introductory synchronise afterward upgrading will be goosy
OneShot	(1) 18 year old who abandoned her child in a hospital later got custody	(1) 18 yr old who untenanted her tyke in a hospital subsequently got detention	(1) 18 class old who deserted her tyke in a hospital subsequently got detention
	(2) the first sync after upgrading will be slow	(2) the eightieth sync later upgrading bequeath be tedious	(2) the eightieth sync later upgrading bequeath be tedious

Table 3: Example input and transformed sentences with varying control values and system variants

表 3 给出了几个随机选出的例子，它们是 3 个不同版本系统输入带有不同控制水平值的不同输入句输出的。我们可以清楚地看到输入内容是基于控制水平值被程序修改的。带 CtrlWithPredictor 行展示了从*中等*到*高级*的可读性等级，表 1 中的可读性得分也佐证了这一点。此外，CtrlNoPredicator 正如预期的那样，由于缺乏明确的反馈，模型无法捕捉控制水平的改变。由于缺乏转换知识，CtrlOneShot 系统在带控制信号的情况下也没有表现出多少改进，如果实施了带有迭代探索/开发的训练，它的表现将不仅如此。我们相信，以上发现为 **RQ1**、**RQ2** 和 **RQ3** 给出了积极的正面回答。

6.2 在反向简化任务上与有监督系统的对比

表 2 的数据表明，所生成复杂句子可以与反向文本简化任务的复杂参考文本一较高下。在该任务下，BLEU 值(Papineni 等人，2002)是常用的衡量指标。我们把有监督 Seq2Seq 作为 skyline，且它取得了比本系统更高的 BLEU 值。然而，对本系统来说 BLEU 值不是很重要，因为我们的系统是无监督的，且设计初衷也不是为了尽可能产生与参考文本重叠的输出文本。必须指出，本系统的其他版本平均可读性得分要比基准语料(平均值 0.50)和 Seq2Seq 都高，这说明我们的系统拥有将输入文本转换得为更为正式的能力。此外，更高的语义相关性得分表明本模型的确能够更好地保留原始语义。

Mueller 等人的系统的亮点是变分自编码器(VAE)和效果预测模块，该模块对输入进行修正使输出具有更好的预期效果。这种方法的问题在于，它局限于从以训练数据形式提供的句子的已知分布中进行生成。如表 1 和表 2 所示，系统性能下降可能是由于训练数据量较少，并且他们的系统没有我们这样的迭代训练 Schemes。上述比较研究为 **RQ4** 提供了思路。

7. 结论与展望

我们提出了一个全新的无监督可控文本转换 NLG 框架，该框架基于现成的语言处理来获得文本转换知识。我们的系统通过提高可读性等级来实现输入文本的正式化，且其可读性等级是可控的。在通用领域数据集上的实验从数量和质量两方面证明了输出的被转换文本的优良品质。系统还学到了重视用户的控制输入并据之做出对应响应。当前方法的一个缺点是，

由于采样策略和 Flesch Kincaid 可读性度量是高度词法化的，因此它只执行词法层面的正式化。我们未来的计划包括探索更好的采样技术来生成复杂的结构和语义的生成句，以及在文本简化和文本风格转换任务下测试该框架。

参考文献:

不想翻了,去原文看看吧