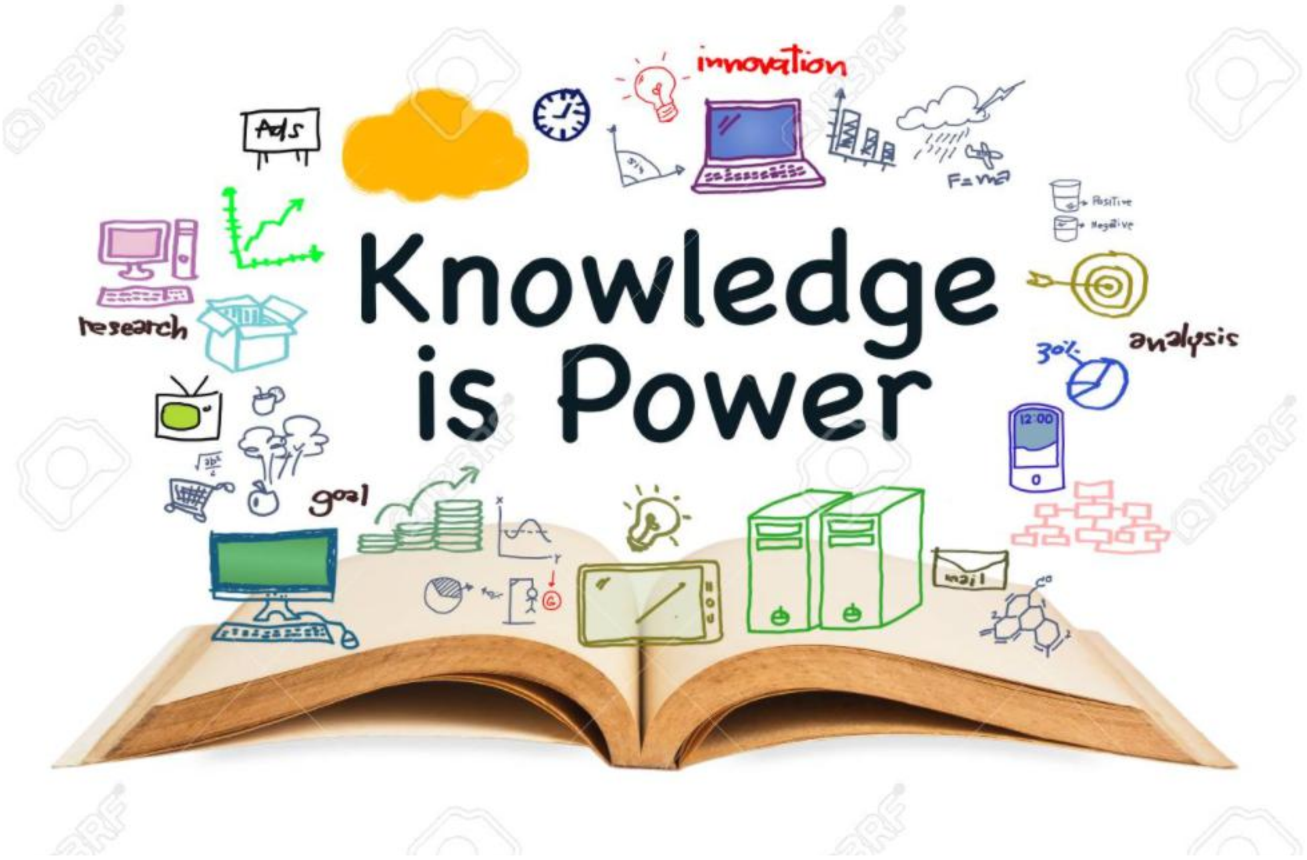
知识增强的文本生成



Knowledge
is Power
innovation
research
analysis
goal
Ads
mail
F=ma

- 在文本生成任务中，“知识”是对输入文本和上下文的一种“补充”，可以由不同方法和信息源获得，包括但不限于关键词，主题，语言学特征，知识库，知识图谱等，这些“知识”可以通过不同的表示方法学习到有效的知识表示，用于增强文本生成任务的生成效果，这就被称为知识增强的文本生成（Knowledge-Enhanced Text Generation）。
- 定义:给定一个文本生成问题，其中系统得到一个输入序列� ，其目的是生成一个输出序列� ，假设我们还可以访问表示为� 的额外知识。知识增强文本生成的目标

- 知识增强的文本生成主要有两个难点:
  - 如何获取有用的知识
  - 如何理解并借助知识促进文本生成

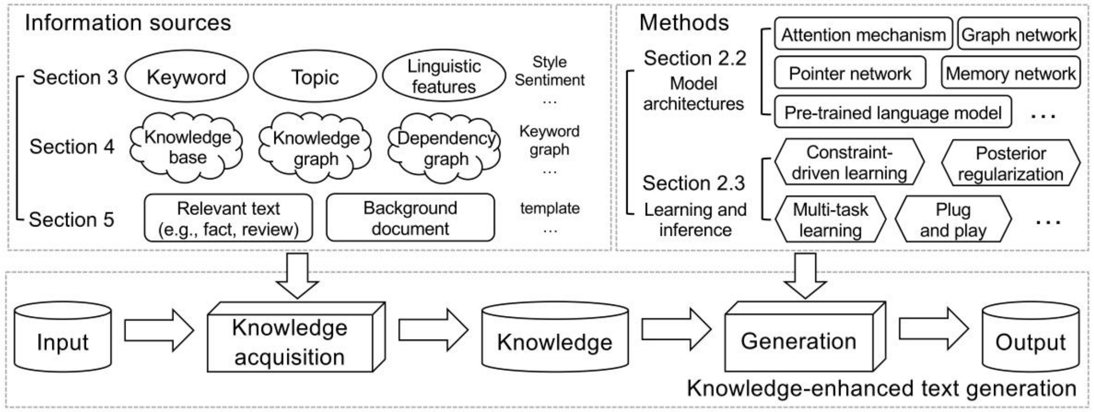


Fig. 1. Categorization of information sources and methods used in knowledge-enhanced text generation systems. Knowledge can be learnt from various information sources, and then integrated into the generation process by a number of methods. Information sources and methods are not limited to the ones listed above.

模型

- 基础文本生成模型
  - 传统文本生成基于encoder-decoder框架 $P(Y|X) = P(y_1, \cdots, y_m|x_1, \cdots, x_n) = \prod_{t=1}^{m} p(y_t|X, y_1, \cdots, y_{t-1}).$
  - RNN-Seq2Seq

$$\mathbf{h}_i = f_{rnn\text{-}en}(\mathbf{e}(x_i), \mathbf{h}_{i-1}), \quad \mathbf{s}_t = f_{rnn\text{-}de}(\mathbf{s}_{t-1}, \mathbf{e}(y_{t-1}), \mathbf{c}),$$
$$p(y_t|y_{t-1}, y_{t-2}, \cdots, y_1, \mathbf{c}) = f_{mlp}(\mathbf{s}_t, \mathbf{e}(y_{t-1}), \mathbf{c}),$$

  - Transformer

$$(\mathbf{h}_1, \mathbf{h}_2, \cdots, \mathbf{h}_n) = f_{tf\text{-}en}(\mathbf{e}(x_1), \mathbf{e}(x_2), \cdots, \mathbf{e}(x_n))$$
$$\mathbf{S}_t = f_{tf\text{-}de}(\mathbf{S}_{t-1}, \mathbf{e}(y_{t-1}), \mathbf{H}),$$

  - 优化：负对数似然损失，目标函数是最大化似然估计MLE

$$\mathcal{L}_{NLL}(\theta) = -\log p_\theta(Y|X) = -\sum_{t=1}^{m} \log\left(p_\theta(y_t|y_{<t}, X)\right).$$

知识融合方法

- **注意力机制**

$$\mathbf{s}_t = f_{de}(\mathbf{s}_{t-1}, \mathbf{e}(y_{t-1}), \mathbf{c}_t).$$

$$\mathbf{c}_t = \sum_{i=1}^{n} \alpha_{ti}\mathbf{h}_i, \text{ where } \alpha_{ti} = \frac{\exp(\eta(\mathbf{s}_{t-1}, \mathbf{h}_i))}{\sum_{k=1}^{n}\exp(\eta(\mathbf{s}_{t-1}, \mathbf{h}_k))},$$

$$\tilde{\mathbf{c}}_t = f_{mlp}(\mathbf{c}_t \oplus \mathbf{c}_t^K)$$

- **拷贝机制**

copynet

$$p_g(y_t) = \begin{cases} \frac{1}{Z}\exp\psi_g(y_t), & y_t \in \mathcal{V} \cup \{\text{unk}\}, \\ 0, & \text{otherwise}; \end{cases}$$

$$p_c(y_t) = \begin{cases} \frac{1}{Z}\sum_{j:x_j=y_t}\exp\psi_c(x_j), & y_t \in \mathcal{V}_X, \\ 0, & \text{otherwise}; \end{cases}$$

$$\begin{aligned} \psi_g(y_t = v_i) &= \mathbf{v}_i^\top \mathbf{W}_g \mathbf{s}_t, \; v_i \in \mathcal{V} \cup \{\text{unk}\}, \\ \psi_c(y_t = x_j) &= \mathbf{h}_j^\top \mathbf{W}_c \mathbf{s}_t, \; x_j \in \mathcal{V}_X, \end{aligned}$$

pointer-generator network

$$p(y_t) = p_m(\text{g}) \cdot p_g(y_t) + (1 - p_m(\text{g})) \cdot p_c(y_t),$$

$$p_m(\text{g}) = \text{sigmoid}(\mathbf{W}_h \cdot \sum_{j=1}^{n}\alpha_{tj}\mathbf{h}_j + \mathbf{W}_s \cdot \mathbf{s}_t + \mathbf{W}_y \cdot \mathbf{e}(y_{t-1})),$$

$$p_g(y_t) = \begin{cases} \frac{1}{Z}\psi_g(y_t), & y_t \in \mathcal{V} \cup \{\text{unk}\}, \\ 0, & \text{otherwise}; \end{cases}$$

$$p_c(y_t) = \begin{cases} \frac{1}{Z}\sum_{j:x_j=y_t}\alpha_{tj}, & y_t \in \mathcal{V}_X, \\ 0, & \text{otherwise}. \end{cases}$$

- **记忆网络**

$$\mathbf{p}_i^k = \text{softmax}((\mathbf{h}_X^k)^\top \mathbf{C}_i^k), \quad \mathbf{o}^k = \sum_i \mathbf{p}_i^k \mathbf{C}_i^{k+1}. \quad \boldsymbol{h}_X^{k+1} = \boldsymbol{h}_X^k + \boldsymbol{o}^k$$

- **图网络**

$$\mathbf{u}^{(k)} = \textsc{Combine}_k(\mathbf{u}^{(k-1)}, \textsc{Aggregate}_k(\{(\mathbf{u}_i^{(k-1)}, \mathbf{e}_{ij}^{(k-1)}, \mathbf{u}_j^{(k-1)}) : \forall(u_i, e_{ij}, u_j) \in \mathcal{N}(u)\})),$$

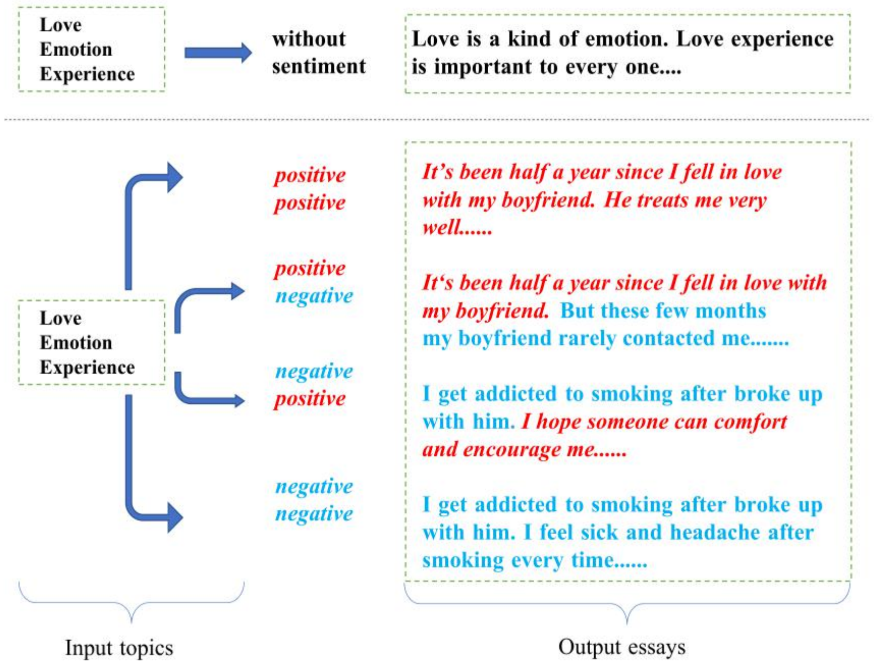
- **预训练模型**

基于主题/关键词/语言学特征

- 主题（Topic）可以看做是文本语义的高度压缩，可以用来保证文本的语义连贯性
    - 借助生成式主题模型中的主题词
    - 同时优化生成模型和卷积主题模型
    - 使用变分推断，利用神经主题模型增强文本生成
- 关键词（Keyword）通常是指一个或多个词，是关于文本重要内容的提炼。
    - 借助关键词分配：从一个可控的词表或预定义的分类系统中选择关键词
    - 借助关键词抽取：从文档中抽取出最具有代表性的单词（e.g.TF-IDF, TextRank, PMI）
- 语言学特征：
    - lemmas
    - part-of-speech (POS) tags
    - dependency parsing
    - semantic parsing。

A Sentiment-Controllable Topic-to-Essay Generator with Topic Knowledge Graph, EMNLP Findings 2020

- 任务：Topic to Essay
- Motivation
  - 加入情感使生成的文本更加多样性且迷人
  - 原有结合知识的工作忽略了知识库的图结构，它只指知识图中的概念，而没有考虑它们的相关性。这种限制导致了概念的相互隔离。
- 方法：
  - 首先将情感信息注入到生成器中，用于控制每个句子的情感，从而生成各种各样的文章。
  - 设计了一个主题知识图增强解码器。与单独使用知识实体的现有模型不同
  - 我们的模型将知识图视为一个整体，并在图中编码更结构化、更关联的语义信息，以生成更相关的文章。



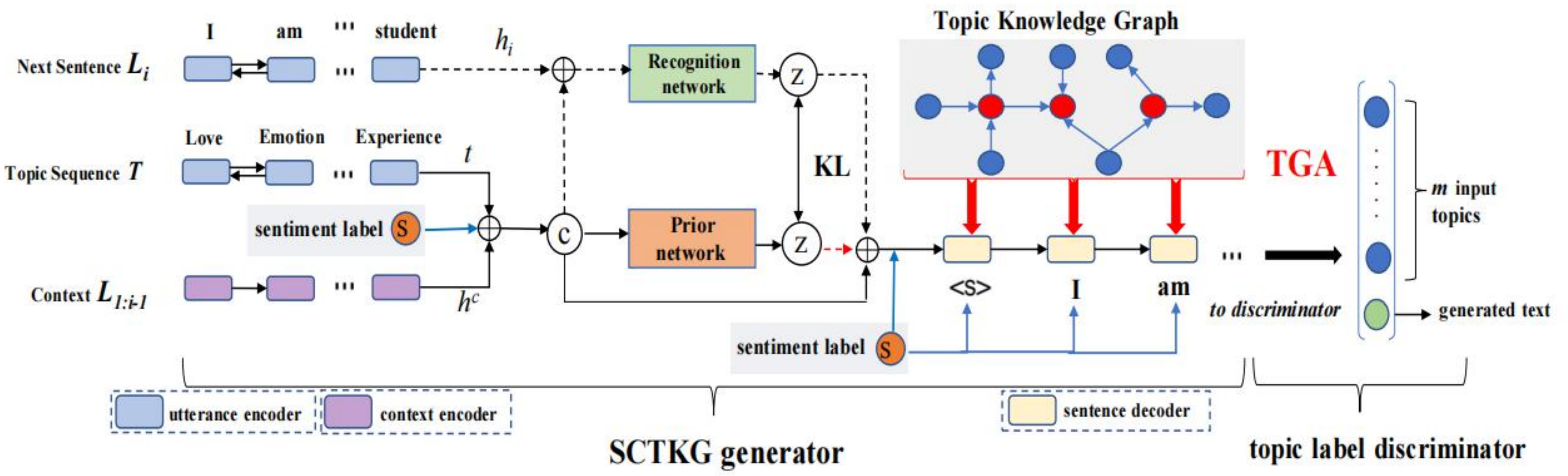
SCTKG Generator：基于CVAE架构

- Encoder：bi-GRU编码主题序列、情感、上下文，计算得到条件变量c= $[e(s);h^c;h^x]$.
  - 后验网络 $q_\phi(z|h_i,c)\sim\mathcal{N}\left(\mu,\sigma^2\mathbf{I}\right)$ $\left[\mu,\sigma^2\right]=\mathrm{MLP}_{\text{recognition}}\left(h_i,c\right),$
  - 先验网络 $p_\theta(z|c)\sim\mathcal{N}\left(\mu',\sigma'^2\mathbf{I}\right)$ $\left[\mu',\sigma'^2\right]=\mathrm{MLP}_{\text{prior}}\left(c\right).$
- Decoder：单层GRU,初始输入 $W_d[z,c,e(s)]+b_d.$
  - 查询向量 $\boldsymbol{q}=[d_{t-1};c;z],$
  - **主题图注意力**将检索到的图中每个三元组与q之间的相关分数。使用相关分数来计算主题词的所有相邻概念的加权和，以形成最终的图向量g_t。
  - 生成一个单词的最终概率 $\mathcal{P}_t=\mathrm{softmax}\left(W_o\left[d_t;e(s);g_t\right]+b_o\right)$

$$g_t=\sum_{n=1}^{N}\alpha_n\boldsymbol{o}_n$$

$$\alpha_n=\frac{\exp\left(\beta_n\right)}{\sum_{j=1}^{N}\exp\left(\beta_j\right)}$$

$$\beta_n=\begin{cases}\left(\mathbf{W}_1\boldsymbol{q}\right)^\top\tanh\left(\mathbf{W}_2\boldsymbol{r}_n+\mathbf{W}_3\boldsymbol{o}_n\right)\\ \text{when}\qquad\qquad\qquad\boldsymbol{o}_n\in\mathcal{S}_1\\ \left(\mathbf{W}_1\boldsymbol{q}\right)^\top\tanh\left(\mathbf{W}_2\boldsymbol{r}_n+\mathbf{W}_4\boldsymbol{o}_n\right)\\ \text{when}\qquad\qquad\qquad\boldsymbol{o}_n\in\mathcal{S}_2\end{cases}$$



- **Topic Label Discriminator**
  - 生成的文本应该与主题词密切相关
  - CNN分类器
  - 假设共有m个主题，鉴别器在（m+1）个类别上生成sigmoid分布，第m+1个索引位置代表样本是生成的文本的概率，，前m个代表是对应主题的概率。
- **训练：两阶段**
  - 第一阶段:类似于传统的CVAE模型，SCTKG Generator的损失$logp(Y|c)$可以表示为:

$$
\begin{aligned}
&-\mathcal{L}(\theta;\phi;c;Y)_{cvae}=\mathcal{L}_{\mathrm{KL}}+\mathcal{L}_{\mathrm{decoder}}\\
&=\mathrm{KL}\left(q_\phi(z|Y,c)\|p_\theta(z|c)\right)\\
&-\mathbb{E}_{q_\phi(z|Y,c)}\left(\log p_{\mathrm{D}}(Y|z,c)\right).
\end{aligned}
$$

  - 第二阶段:生成器和主题标签鉴别器之间采用了对抗性训练。

Experiment

	Automatic evaluation					Human evaluation			
Methods	**BLEU**	**Consistency**	**Novelty**	**Dist-1**	**Dist-2**	**Con.**	**Nov.**	**E-div.**	**Flu.**
TAV	6.05	16.59	70.32	2.69	14.25	2.32	2.19	2.58	2.76
TAT	6.32	9.19	68.77	2.25	12.17	1.76	2.07	2.32	2.93
MTA	7.09	25.73	70.68	2.24	11.70	3.14	2.87	2.17	3.25
CTEG	9.72	39.42	75.71	5.19	20.49	3.74	3.34	3.08	3.59
SCTKG(w/o-Senti)	9.97	**43.84**	78.32	5.73	**23.16**	**3.89**	3.35	3.90	3.71
SCTKG(Ran-Senti)	9.64	41.89	**79.54**	5.84	23.10	3.80	**3.48**	**4.29**	3.67
SCTKG(Gold-Senti)	**11.02**	42.57	78.87	**5.92**	23.07	3.81	3.37	3.94	**3.75**

Methods	**Precison**	**Recall**	**Senti-F1**
Full model	**0.68**	**0.66**	**0.67**
w/o Enc-senti	0.56	0.55	0.56
w/o Dec-senti	0.59	0.62	0.61
w/o TGA	0.62	0.64	0.63

Input topics: Law Education
Sentiment label: neu. pos. neg. neg. neu.
Output essay: I am a senior high school student. I am in the best high school in our town. But bullying still exist on our campus. Teachers always ignore this phenomenon. What should we do to protect our rights?

基于知识库、知识图谱、依赖图增强

- 知识库（knowledge base, KB）就是一个收集、存储和管理大规模知识信息的技术。知识库包含大量的三元组（triple），三元组由subjects，predicates和objects组成，可以被称为事实（facts）或事实三元组（factual triples）。
  - 针对知识库设计有监督任务进行联合优化
  - 针对知识库设计无监督方法，作为额外条件因素
  - 选择知识库或事实增强知识融合
- 知识图谱：
  - 将知识图谱嵌入融入文本生成
  - 通过路径寻找策略在知识图谱上进行推理
  - 使用图神经网络增强图表示
- 依赖图增强的文本生成
  - 语法依赖图：有向无环图，表示单词之间的语法关系。
  - 语义依赖图：语义依赖图上的节点可以通过semantic role labeling （SRL）和dependency parsing抽取得到，然后节点可以通过不同的关系进行连接

Language Generation with Multi-Hop Reasoning on Commonsense Knowledge Graph，EMNLP 2020

- 背景
  - 语言模型通过在大量语料上预训练**隐式**地学习到了一定的知识，但没有**显式**利用知识库和知识图谱，对知识的结合较为低效且不系统
  - 融合外部知识增强模型推理能力的研究仅仅依靠独立的知识三元组，忽略了知识图谱中知识之间的丰富相关性，这些相关性可能为复杂的推理提供多个合理的证据
- 目的
  - 基于常识知识图谱的推理，完成语言生成任务。

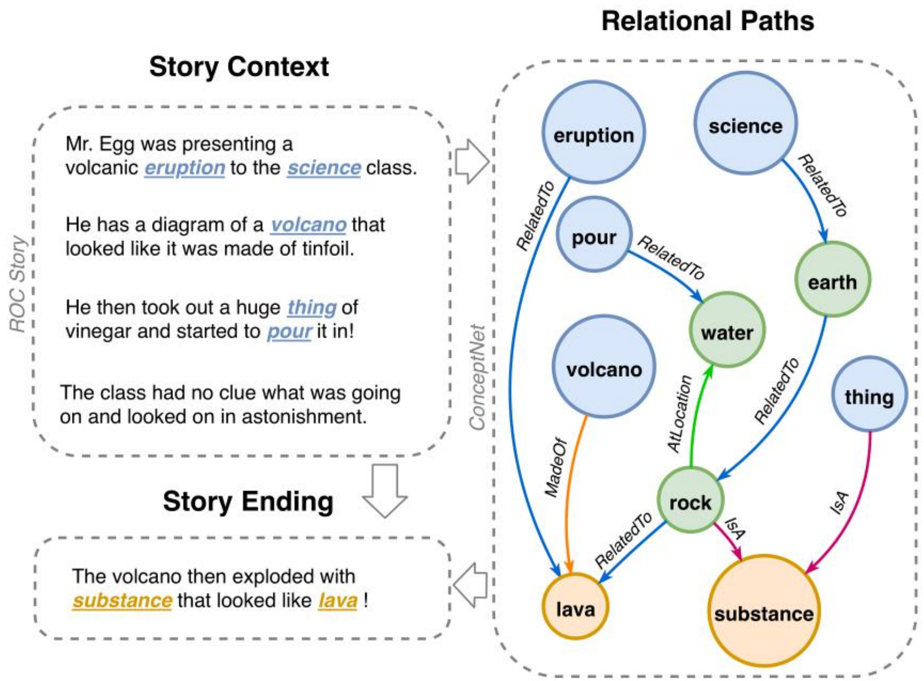


Figure 1: An example of using structural relational knowledge as commonsense grounding in story ending generation. Blue nodes correspond to the concepts in the context, orange nodes correspond to those in the story ending and gree nodes are intermediate concepts that connect the evidence chain.

- 静态多关系图编码:用非参数操作φ(·)对多关系图进行编码，以组合关系和概念。

$$o_v^l = \frac{1}{|\mathcal{N}(v)|} \sum_{(u,r)\in\mathcal{N}(v)} \mathbf{W}_N^l \phi(\boldsymbol{h}_u^l, \boldsymbol{h}_r^l), \qquad \boldsymbol{h}_r^{l+1} = \mathbf{W}_R^l \boldsymbol{h}_r^l.$$

$$\boldsymbol{h}_v^{l+1} = \mathrm{ReLU}\left(\boldsymbol{o}_v^l + \mathbf{W}_S^l \boldsymbol{h}_v^l\right),$$

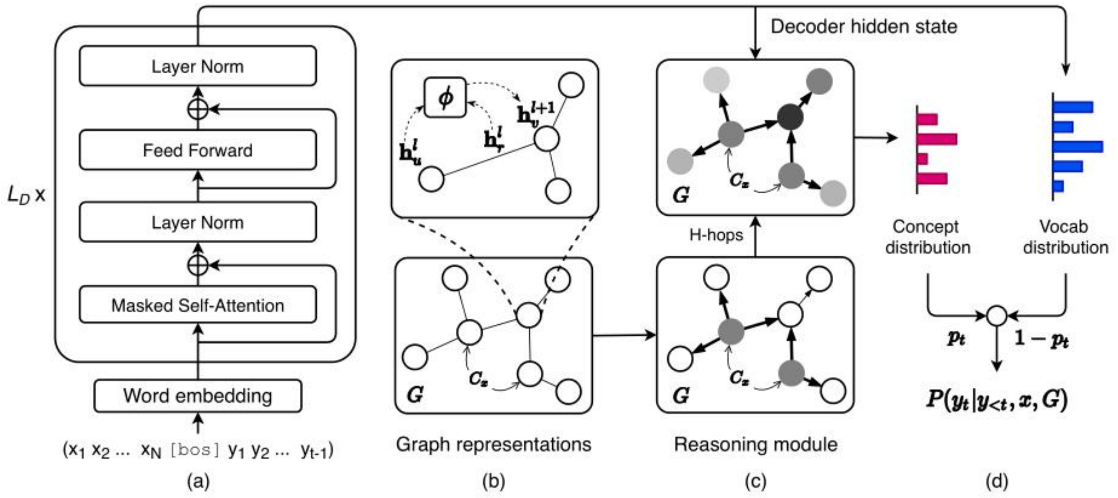
- 预训练Transformer上下文建模：GPT-2

$$\boldsymbol{h}_t^0 = \boldsymbol{e}_t + \boldsymbol{p}_t,$$
$$\boldsymbol{h}_t^l = \mathtt{T\_block}(\mathbf{H}_{\leq t}^{l-1}), l \in [1, L_D]$$
$$P(s_t|\boldsymbol{s}_{<t}) = \mathrm{softmax}(\mathbf{W}_{LM}\boldsymbol{h}_t^{L_D} + \boldsymbol{b})$$

- 门控生成最终分布：
  - 知识图谱的概念分布和仅基于输入语料的标准词典分布的加权分布作为最终的输出分布，确定next token

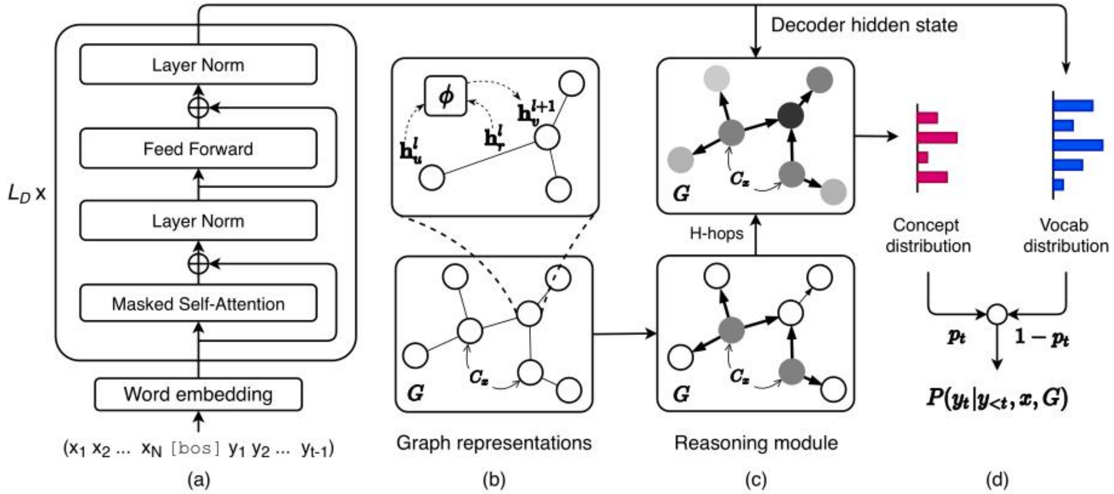
$$g_t = \sigma\left(\mathbf{W}_{gate}\boldsymbol{h}_t^{L_D}\right)$$

$$P(y_t|\boldsymbol{y}_{<t}, \boldsymbol{x}, G) = g_{t+N} \cdot P(c_{t+N}|\boldsymbol{s}_{<t+N}, G) + (1 - g_{t+N}) \cdot P(s_{t+N}|\boldsymbol{s}_{<t+N})$$



- 训练：
  - 最小化NLL $\mathcal{L}_{gen} = \sum_{t=1}^{M+1} -\log P(y_t^{\text{gold}}|\boldsymbol{y}_{<t}^{\text{gold}}, \boldsymbol{x}, G).$
  - 辅助损失：鼓励模型更有效地学习图上的多跳推理
    - L_{gate}监督选择一个概念或一个通用词的概率
    - L_{weak}使预测的三元组相关性匹配从BFS得到的边的启发式标签到图上y^{gold}中的概念。
  - 最终优化函数：$\mathcal{L}_{gen} + \alpha\mathcal{L}_{gate} + \beta\mathcal{L}_{weak}$

- 多跳推理流生成
  - 在生成过程中对图结构进行显式推理，设计了一个动态推理模块，在每一个解码步骤中利用知识图的结构模式和上下文信息沿关系路径传播证据。
  - 首先，初始化知识图谱G各个节点的分值，模块通过多次更新外部节点与被访问的邻居的得分来广播G上的信息，直到访问了G上的所有节点。最初，Cx中对应概念的节点被赋予1分，其他未访问的节点被赋予0分。



$$ns(v) = \underset{(u,r)\in\mathcal{N}_{in}(v)}{f}\left(\gamma \cdot ns(u) + R(u,r,v)\right)$$

  - 然后，通过关系路径的多跳推理更新节点分值。对于知识图谱G，利用已经访问过的节点来更新未访问的邻接节点的得分，多跳直到访问完G中所有节点，实现对所有节点的得分更新。对于未访问的节点$v \in V$，它的节点得分$ns(v)$是通过聚集来自的证据来计算的$N_{in}(v)$表示直接连接v的已访问节点u及其边r的集合。

$$R(u,r,v) = \sigma\left(\boldsymbol{h}_{u,r,v}^{\mathrm{T}} \mathbf{W}_{\mathrm{sim}} \boldsymbol{h}_t^{L_D}\right)$$
$$\boldsymbol{h}_{u,r,v} = \left[\boldsymbol{h}_u^{L_G}; \boldsymbol{h}_r^{L_G}; \boldsymbol{h}_v^{L_G}\right]$$

  - 最后，对所有节点的得分做softmax归一化，得到最终的概念分布

$$P(c_t|\boldsymbol{s}_{<t}, G) = \mathrm{softmax}_{v\in V}(ns(v))$$

实验

- 提取知识子图
  - 从G中提取一个子图G = (V,E)，它由多个相互连接的路径组成，从输入序列中的源概念Cx开始。为了从输入文本序列中识别概念，使用Spacy3对表层文本的lemmatized形式进行模糊匹配，并过滤掉停止词。跟只考虑动词和名词作为候选概念，因为发现提取的图与所有匹配的概念相比噪声更大。
  - 具体地说，对以下过程进行迭代H跳:
    - 从当前子图的节点开始(Cx初始化)，搜索每个节点的直接邻居，保留有连通边的topB节点，扩大子图。
    - 对于每个候选节点，根据其进入该节点的程度进行选择。候选节点v的进入度定义为当前子图中与v直接连接的节点的数量。
    - 直观上，保留了那些经常访问的节点和支持图上信息流的突出概念。
- 任务
  - 故事结尾生成(SEG) 是为一个四句话的故事上下文生成一个合理的结尾。
  - 归纳法自然语言生成(NLG) 是生成一个解释性假说给出两个观察:O1作为原因和O2作为后果。
  - 解释生成(EG) 是在给出一个反事实陈述的情况下生成一个解释，以便进行意义构建。

结果与分析

Graph statistics	EG	αNLG	SEG
Avg. # Concepts	193.1	201.6	208.5
Avg. # Triples	1094.3	1324.6	1148.6

- 自动评估

Models	EG				αNLG			
	BLEU-4	METEOR	ROUGE-L	CIDEr	BLEU-4	METEOR	ROUGE-L	CIDEr
Seq2Seq	6.09	24.94	26.37	32.37	2.37	14.76	22.03	29.09
COMeT-Txt-GPT2	N/A	N/A	N/A	N/A	2.73†	18.32†	24.39†	32.78†
COMeT-Emb-GPT2	N/A	N/A	N/A	N/A	3.66†	19.53†	24.92†	32.67†
GPT2-FT	15.63	38.76	37.32	77.09	9.80	25.82	32.90	57.52
GPT2-OMCS-FT	15.55	38.28	37.53	75.60	9.62	25.83	32.88	57.50
GRF	**17.19**	**39.15**	**38.10**	**81.71**	**11.62**	**27.76**	**34.62**	**63.76**

Table 3: Automatic evaluation results on the test set of EG and αNLG. Entries with N/A mean the baseline is not designated for this task. †: we use the generation results from Bhagavatula et al. (2020).

Models	BLEU-1/2	Distinct-2/3
Seq2Seq	19.1 / 5.5	0.181 / 0.360
IE+GA	20.8 / 6.4	0.140 / 0.280
WriterForcing	16.5 / 3.7	0.335 / 0.584
GPT2-FT	25.5 / 10.2	0.304 / 0.505
GPT2-OMCS-FT	25.5 / 10.4	0.352 / 0.589
GRF	**26.1 / 11.0**	**0.378 / 0.622**

Table 4: Automatic evaluation on the test set of SEG.

- 人工评估

Models	EG				αNLG				SEG			
	Fluency		Reasonability		Fluency		Reasonability		Fluency		Reasonability	
	W	L	W	L	W	L	W	L	W	L	W	L
vs. IE+GA	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	0.62**	0.07	0.72**	0.11
vs. COMeT-Emb-GPT2	N/A	N/A	N/A	N/A	0.31**	0.14	0.55**	0.25**	N/A	N/A	N/A	N/A
vs. GPT2-FT	0.24**	0.09	0.54**	0.21	0.15*	0.10	0.56**	0.20	0.21**	0.12	0.45**	0.19
vs. GPT2-OMCS-FT	0.18**	0.09	0.58**	0.18	0.12	0.09	0.50**	0.20	0.17*	0.11	0.40**	0.15

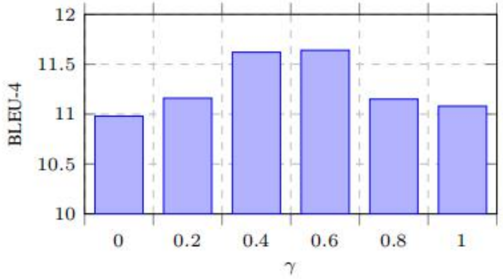
Criteria	EG	αNLG	SEG
Fluency	0.615	0.543	0.315
Reasonability	0.551	0.677	0.595

- 消融研究

Models	BLEU-4	ROUGE-L
GRF	11.62	34.62
w/ `mean(·)` aggregator	11.32	34.46
w/o DMRF	10.67	33.75
w/o SMGE	11.10	34.36

Table 5: Ablation study on the test set of αNLG. SMGE denotes static multi-relational graph encoding (see §3.2.1) and DMRF denotes dynamic multi-hop reasoning flow (see §3.2.3).

动态多跳推理的有效性

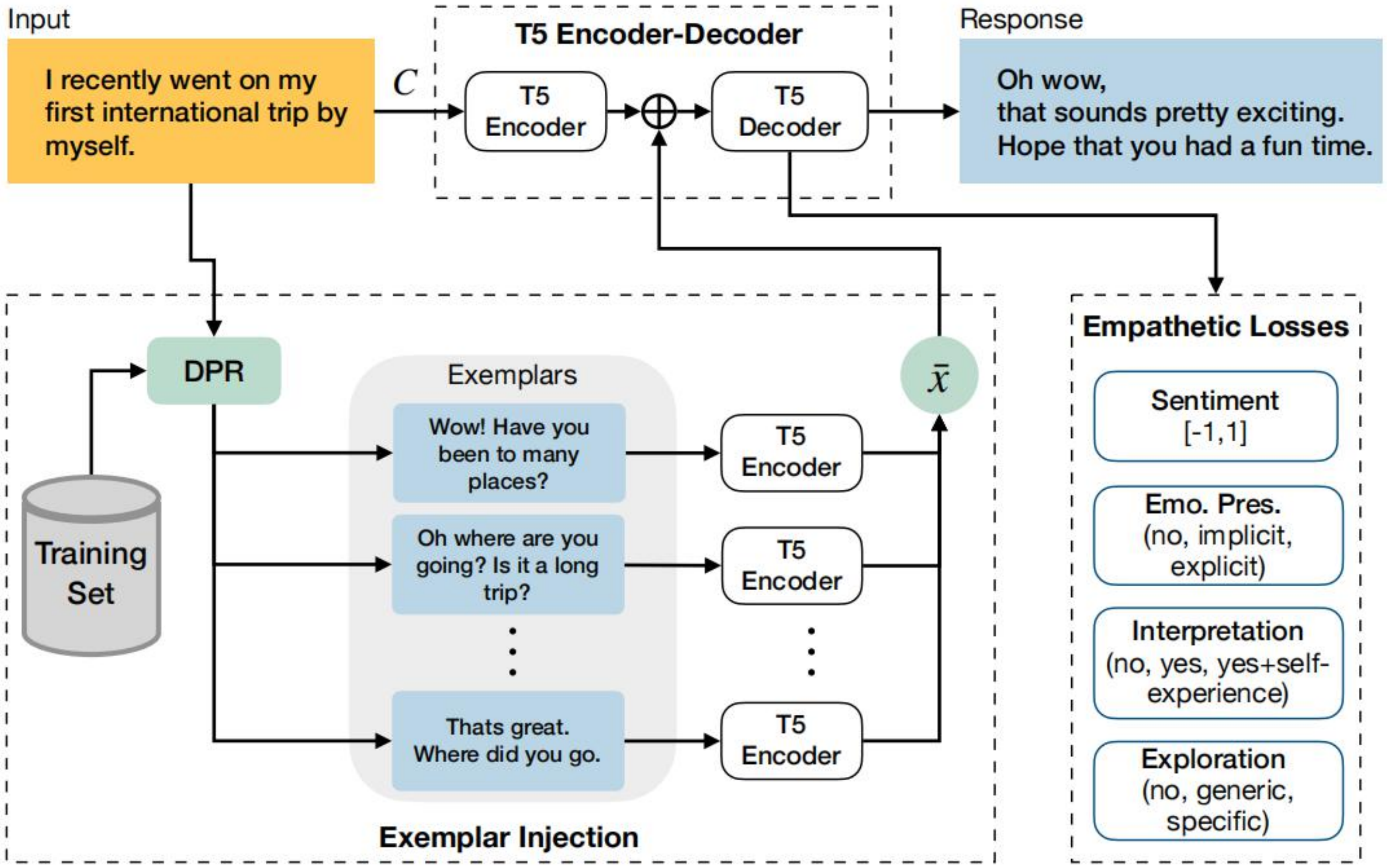


基于背景文本增强

- 这里的文本指的是与输入序列相关，能够提供额外知识的文本，称为knowledge grounded text
- 这些文本不会出现在训练语料和结构化库（例如知识库和知识图谱）中，但是可以在线资源中获取大量相关文本，这些文本对于理解文本语义和上下文语境发挥重要的作用，可以作为重要的知识信息帮助文本生成任务
  - 百科全书文章可以提供关于输入文本的解释和背景信息
  - 购物网站包含了许多商品相关问题的答案和意见
  - 社交媒体提供了关于某个事件人们的看法和观点
- 用检索到的信息指导生成。
- 将背景文档建模到回复生成。

Exemplars-guided Empathetic Response Generation Controlled by the Elements of Human Communication，CIKM 2021

- 采用exemplars范例的想法进行可控文本生成
- 其中从训练集中检索一组范例响应，语义上与输入上下文相关，并作为模板馈送到响应生成器。
- 这些模板响应通过响应上的风格和主题提示来引导生成器，这些提示被认为是对用户的移情。
- 这类似于为解决一个困难的数学问题提供提示，以显著缩小搜索空间。类似地，我们推测范例可以限制解码器的搜索空间。



方法

- Encoder-Decoder Transformer
  - 连接上下文话语 $C = [u_1 \oplus u_2 \oplus \cdots \oplus u_n] = [w_{11}, w_{12}, \ldots, w_{nm}]$
  - 词嵌入和说话人嵌入之和 $E(C) = E_W(C) + E_S(C)$

$$Z = \mathrm{T5}_{enc}^{ctx}(E(C)),$$
$$\mathcal{P}_{resp} = \mathrm{T5}_{dec}(E_W(R_{1:t-1}), Z),$$

- 样例抽取：DPR $p(\Psi|C) = \exp(E_{cand}(\Psi)^T E_{query}(C)),$
- 注入：
  - 每个范例用T5 encoder编码 $\mathcal{Z}_i = \mathrm{T5}_{enc}^{exl}(E_W(\Psi_i)).$
  - 平均池化 得到token级和vector级样例表示

    $\psi_i = \mathrm{mean}(\mathcal{Z}_i) \in \mathbb{R}^{n_{emb}}$ $\chi = \mathrm{mean}([\psi_1, \psi_2, \ldots, \psi_q]).$
  - 与context融合 $Z_{fused} = \mathrm{FC}_{exl}([Z_i \oplus \chi]_{i=1}^k).$
  - 使用融合表示生成response $\mathcal{P}_{resp} = \mathrm{T5}_{dec}(E_W(R_{1:t-1}), Z_{fused}),$
  - 损失 $L_{gen} = -\log p(R_{gold}|C),$ $p(R_i|C, R_{0:i-1}) = \mathcal{P}_{resp}[i].$

- 回复的共情损失
  - 情绪存在Emotional Presence：response中是否存在情绪（不存在，隐式存在，显式存在）
  - 解释Interpretation：回复是否能正确解释用户
  - 探索Exploration:回复是否尝试探索用户的情绪（无，一般的，具体的）
  - 情感Sentiment：[-1,1]两级分别是消极与积极
  - 前三者使用对应数据集，按1/2/3分训练分类器，计算交叉熵损失
  - sentiment计算MSE损失 $\mathcal{P}_* = \mathrm{softmax}(\mathrm{FC}_*(\tilde{R}[0])).$
- Training

$$\mathcal{L} = \alpha_{gen}L_{gen} + \alpha_{EP}L_{EP} + \alpha_{int}L_{int} + \alpha_{exp}L_{exp} + \alpha_{sent}L_{sent},$$

训练DPR

- 预训练的DPR模型在基于移情对话训练集的数据集上进行微调。DPR的每个训练样本都由一个输入上下文和一组响应组成，这些响应包含一个正回复和n_{neg}个负回复，它们是从具有**不同**情绪的其他训练对话的话语中随机取样的:

$$\overbrace{[u_1, u_2, \ldots, u_j]}^{\text{Context}}, [\underbrace{R_{gold}}_{\text{Positive Response}} \overbrace{, R_1^-, R_2^-, \ldots, R_{n_{neg}}^-]}^{\text{Negative Responses}}$$

- DPR被微调以从提供的一组响应中预测正确的正样本。它首先使用两个称为双编码器的独立编码器将上下文和响应编码到一个连续的空间中。然后，通过同时最大化上下文和黄金响应之间的相似性以及最小化上下文和否定响应之间的相似性来执行双编码器训练。点积被用作相似性的度量，正响应的负对数似然被优化为目标函数:

$$L_{dpr} = -\log \frac{e^{sim(c, r_{gold})}}{e^{sim(c, r_{gold})} + \sum_{j=1}^{n_{neg}} sim(c, r_j^-)},$$

- 然而，在样本提取的推理过程中，对于每个输入上下文，来自具有**相同**情感的移情对话的训练对话的一组响应被馈送到DPR。基于DPR的点积与上下文相似度最高的top q响应被选为样本。值得注意的是，样本响应集不包含黄金响应或来自训练样本的相同对话的任何其他响应。换句话说，样本反应总是从不同的对话中选出(但属于同一情感类别)。

Algorithm 2: DPR Training Algorithm

Training Dataset $\mathcal{D}$ = EmpatheticDialogues Training Split;
Initialize pre-trained BERT-Base as context encoder **C**;
Initialize pre-trained BERT-Base as response encoder **R**;
Dot product similarity function $sim(x, y) = x.y$;
for *dialogue id* d *with context* c, *response* r *and emotion* e *in* $\mathcal{D}$ **do**
  Assign positive response:
    $r^+ = r$
  Assign negative responses:
    $\mathcal{D}_1 = \mathcal{D} \setminus d$ with emotion=e ;
    $\mathcal{D}_2 = \mathcal{D} \setminus d$ with emotion$\neq e$;
    r_1^- = random sample of responses from $\mathcal{D}_1$;
    r_2^- = random sample of responses from $\mathcal{D}_2$;
    $r^- = r_1^- \cup r_2^-$
  Compute exponentiated similarities:
    $s^+ = exp(sim(\mathrm{C}(c), \mathrm{R}(r^+)))$;
    $s^- = \sum_{j=1}^{n} exp(sim(\mathrm{C}(c), \mathrm{R}(r_j^-)))$;
  Compute negative log-likelihood of positive response:
    $l = -log(\frac{s^+}{s^+ + s^-})$;
  Backpropagate loss l to update **C**, **R**;
end

Algorithm 3: DPR Inference Algorithm

Dataset $\mathcal{D}$ = EmpatheticDialogues Training Split;
Collect all responses $\mathcal{R}$ from $\mathcal{D}$:
  $\mathcal{R} = \{r_1, r_2, .., r_n\}$;
for *dialogue id* d *with context* c *in train/val/test dataset* **do**
  Compute similarities with $\mathcal{R}$:
  for *j=1 to n* **do**
    $s_j = sim(\mathrm{C}(c), \mathrm{R}(r_j))$;
  end
  $s = [s_1, s_2, .., s_j]$;
  Sort s in decreasing order ;
  Return responses corresponding to first k elements of sorted s ;
end

实验

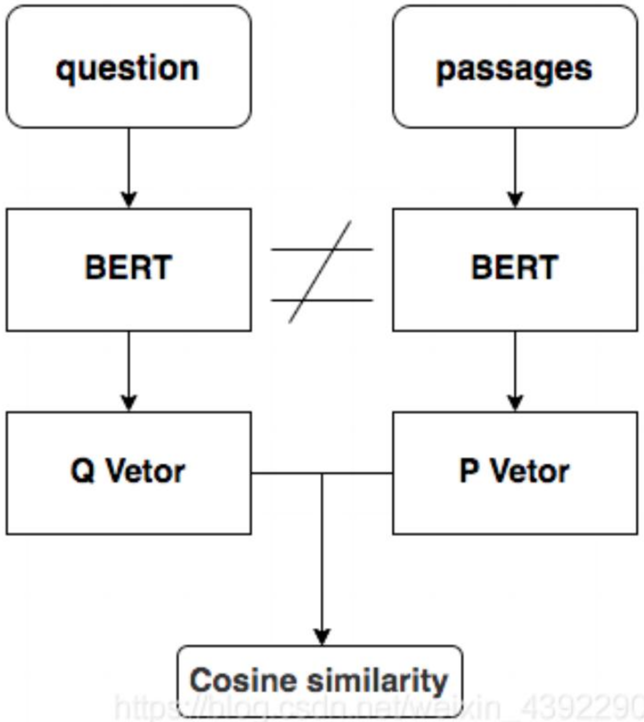
Method	#params.	BLEU	PPL	*Coarse Attributes*			*Empathy-Based Attributes*			*Diversity* (%)	
				Empathy	Relevance	Fluency	Emotion Presence	Interpretation	Exploration	Distinct-1	Distinct-2
Gold		100	–	4.48	4.98	4.98	1.95	1.92	2.34	7.94	42.70
MIME	16.95M	8.76	37.33	3.00	3.15	4.39	1.75	1.4	1.22	0.63	3.97
EmpDG (SOTA)	28.38M	8.61	34.18	2.95	3.1	4.32	1.64	1.3	1.69	**1.81**	6.38
T5-PT	60.50M	7.47	36.88	3.31	3.75	**4.78**	1.61	1.49	1.82	1.59	**17.33**
T5-GloVe/LEMPEx w/o Exemplars, w/o Emp. Losses	27.57M	7.38	**25.26**	3.34	3.36	4.42	1.84	1.43	1.68	1.37	13.72
LEMPEx w/o Emp. Losses	165.33M	7.49	26.88	3.55	3.69	4.65	1.88	1.52	1.76	1.39	14.55
LEMPEx	165.33M	**7.88**	26.37	**3.76**	**3.78**	4.61	**2.02**	**1.64**	**2.02**	1.41	14.66

Context	**Reference**	**Exemplar Fine-Tuned DPR**	**Exemplar Pre-Trained DPR**
I was going through some boxes the other day. I found some old pictures of my kids I thought were gone.	That is exciting! I love having pictures to look back on.	Pictures are nice to have to be able to look back on life.	Those are nice memories.
I recently hit a new personal record at the gym! I was very excited.	That is a great accomplishment. I admire your dedication in obtaining that goal.	Whoa, that is really impressive! How much do you bench now?	Ohh nice!
I live alone. Sometimes it can be hard with nobody around.	That must feel lonely. You should go out and try to meet some people!	I can imagine that. I've never lived alone before.	How are you keeping yourself occupied?
I wish my neighbors were more considerate. They woke me up at 4am recently.	Wow did you call someone about that?	That was rude. Did you confront them?	Oh yeah? Why is that?
We just found out the person working in our accounting department has been stealing money from our company for years.	Yikes! That is terrible! What a awful thing to do! I am glad he finally got busted! Wow, for years, huh?	Wow, that is pretty crazy! What did you do?	I would have screamed!

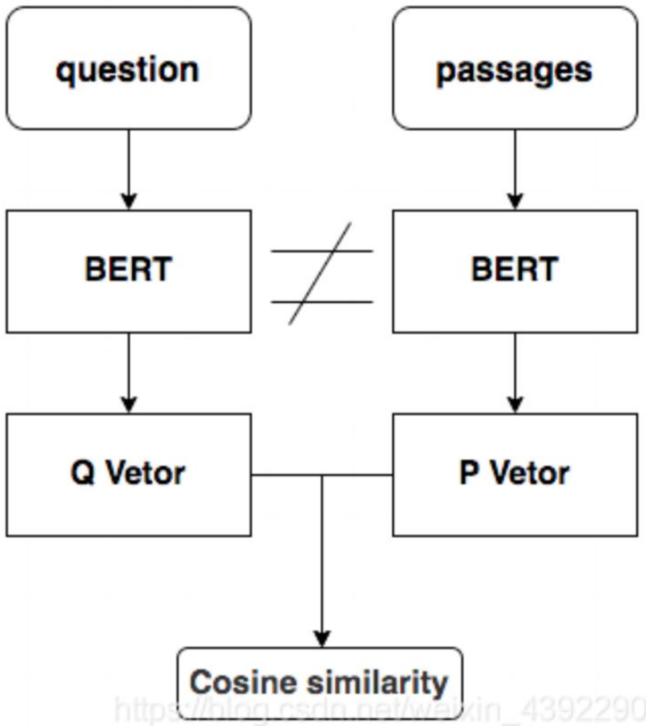
Emotion	ANXIOUS
Context	I'm starting a new job next week, and while I am excited, I am also very nervous! Congratulations! Don't be too nervous! You must have done something right to get the job in the first place! That's true, but the uncertainty always makes me feel a bit nervous.
Reference	That is natural. I'm sure you will do great!
Exemplar	Just take deep breaths and act confidently and you'll do great!
LEMPEx	I hope that you do try to stay confident. You will be fine.
w/o Exemplar	That is awesome! I wish you the best! Hope you get your job!
Emotion	GRATEFUL
Context	My neighbor really helped me out recently after a big windstorm.
Reference	That was very kind of them. Did they charge you for their labor?
Exemplar	That's so nice of him to do. No one wants to help anyone out these days but I'm glad for you.
LEMPEx	That was really nice of them. I'm glad that you had a very good neighbor.
w/o Exemplar	Wow that is very scary, what is the one. My family did that all the
Emotion	FURIOUS
Context	Not too long ago, I took a trip on a plane. The airline lost my whole suitcase!
Reference	Oh man! how long did it take to get your stuff back?
Exemplar	Oh no! Did they reimburse you?
LEMPEx	Did you call the police? They are so careless people.
w/o Exemplar	Oh man! That is terrible! No one happened to me in a car!
Emotion	GRATEFUL
Context	My dad called me one day crying out for help. I had to drive over to his house.
Reference	Oh my goodness! What happened?
Exemplar	Why what happened?
LEMPEx	Oh that sounds scary! Did he say what was going on? Is he alright?
w/o Exemplar	That's awful. Is he okay? Does it have someone you can do that.

Dense Passage Retriever (DPR)，EMNLP 2020

- Motivation:
  - 开放域问题回答依赖于高效的文本检索来选择候选段落，传统的稀疏向量空间模型用的较多的有TF-IDF或 BM25算法，但这些算法仅仅是在词的匹配上进行检索，并未考虑语义的相关性，有很大的局限性
- 解决的问题：提高QA领域中问题段落检索准确性
- **模型结构**
  - 首先，将问题输入BERT得到d维的向量Q，将段落输入另外一个BERT得到d维的向量P
  - 然后将Q和P计算余弦相似度，即可得到两个向量之间的距离
  - 其中问题和段落使用的BERT参数是不共享的。



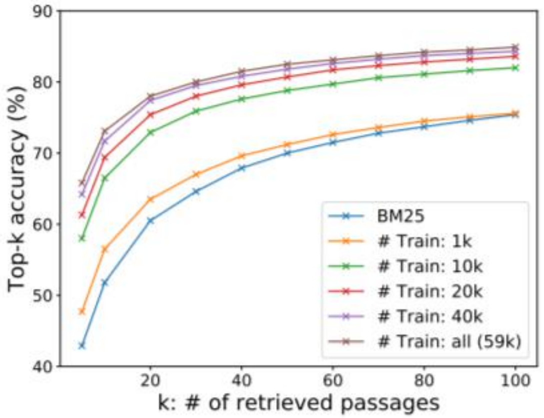
- **训练**
  - 由于原始的BERT预训练模型很通用，而且得到的句向量是不具备相似性计算要求的，也就是说，相似的两句话输入BERT得到的两个向量并不一定很近。
  - 重新训练了一个BERT，专门用来生成问题段落的句向量表示。
  - 数据构造
    - 负样本：
      - Random：从语料中随机抽取；
      - BM25：使用BM25检索的不包含答案的段落；
      - Gold：训练集中其他问题的答案段落
    - 正样本：
      - 通过BM25算法，在Wikipedia中进行检索，取top-100的段落，如果答案没有在里面，则丢掉这个问题。对于 SQuAD 和 NQ数据集，同样使用问题在Wikipedia检索，如果检索到的正样本段落能够在数据集中匹配则留下，否则丢掉。
  - 损失函数



$$L(q_i, p_i^+, p_i^-, \ldots, p_{i,n}^-) = -log \frac{e^{sim(q_i, p_i^+)}}{e^{sim(q_i, p_i^+)} + \sum_{j=1}^{n} e^{sim(q_i, p_{i,j}^-)}}$$

$$\mathrm{sim}(q, p) = E_Q(q)^\intercal E_P(p),$$

实验



Training	Retriever	Top-20 NQ	Top-20 TriviaQA	Top-20 WQ	Top-20 TREC	Top-20 SQuAD	Top-100 NQ	Top-100 TriviaQA	Top-100 WQ	Top-100 TREC	Top-100 SQuAD
None	BM25	59.1	66.9	55.0	70.9	68.8	73.7	76.7	71.1	84.1	80.0
Single	DPR	78.4	79.4	73.2	79.8	63.2	85.4	**85.0**	81.4	89.1	77.2
	BM25 + DPR	76.6	79.8	71.0	85.2	**71.5**	83.8	84.5	80.5	92.7	**81.3**
Multi	DPR	**79.4**	78.8	**75.0**	**89.1**	51.6	**86.0**	84.7	**82.9**	93.9	67.6
	BM25 + DPR	78.0	**79.9**	74.7	88.5	66.2	83.9	84.4	82.3	**94.1**	78.6

Type	#N	IB	Top-5	Top-20	Top-100
Random	7	✗	47.0	64.3	77.8
BM25	7	✗	50.0	63.3	74.8
Gold	7	✗	42.6	63.1	78.3
Gold	7	✓	51.1	69.1	80.8
Gold	31	✓	52.1	70.8	82.1
Gold	127	✓	55.8	73.0	83.1
G.+BM25$^{(1)}$	31+32	✓	65.0	77.3	84.4
G.+BM25$^{(2)}$	31+64	✓	64.5	76.4	84.0
G.+BM25$^{(1)}$	127+128	✓	**65.8**	**78.0**	**84.9**

Training	Model	NQ	TriviaQA	WQ	TREC	SQuAD
Single	BM25+BERT (Lee et al., 2019)	26.5	47.1	17.7	21.3	33.2
Single	ORQA (Lee et al., 2019)	33.3	45.0	36.4	30.1	20.2
Single	HardEM (Min et al., 2019a)	28.1	50.9	-	-	-
Single	GraphRetriever (Min et al., 2019b)	34.5	56.0	36.4	-	-
Single	PathRetriever (Asai et al., 2020)	32.6	-	-	-	**56.5**
Single	$\text{REALM}_{\text{Wiki}}$ (Guu et al., 2020)	39.2	-	40.2	46.8	-
Single	$\text{REALM}_{\text{News}}$ (Guu et al., 2020)	40.4	-	40.7	42.9	-
Single	BM25	32.6	52.4	29.9	24.9	38.1
	DPR	**41.5**	56.8	34.6	25.9	29.8
	BM25+DPR	39.0	57.0	35.2	28.0	36.7
Multi	DPR	**41.5**	56.8	**42.4**	49.4	24.1
	BM25+DPR	38.8	**57.9**	41.1	**50.6**	35.8

Table 4: End-to-end QA (Exact Match) Accuracy. The first block of results are copied from their cited papers.