INJECTING KNOWLEDGE INTO LANGUAGE GENERATION: A CASE STUDY IN AUTO-CHARTING AFTER-VISIT CARE INSTRUCTIONS FROM MEDICAL DIALOGUE

Anonymous authors
Paper under double-blind review

ABSTRACT

Factual correctness is often the limiting factor in practical applications of natural language generation in high-stakes domains such as healthcare. An essential requirement for maintaining factuality is the ability to deal with rare tokens. This paper focuses on rare tokens that appear in both the source and the reference sequences, and which, when missed during generation, decrease the factual correctness of the output text. For high-stake domains that are also knowledge-rich, we show how to use knowledge to (a) identify which rare tokens that appear in both source and reference are important and (b) uplift their conditional probability. We introduce the "utilization rate" that encodes knowledge and serves as a regularizer by maximizing the marginal probability of selected tokens. We present a study in a knowledge-rich domain of healthcare, where we tackle the problem of generating after-visit care instructions based on patient-doctor dialogues. We verify that, in our dataset, specific medical concepts with high utilization rates are underestimated by conventionally trained sequence-to-sequence models. We observe that correcting this with our approach to knowledge injection reduces the uncertainty of the model as well as improves factuality and coherence without negatively impacting fluency. [1]

1 INTRODUCTION

Recent advances in language modeling (*c.f.* Dong et al. (2021); Erdem et al. (2022) for survey) have enabled applications across multiple domains including education (Shen et al., 2021), jurisprudence (Bell et al., 2021), e-commerce (Zhang et al., 2020; Xiao et al., 2021), and healthcare (Valmianski et al., 2021; Compton et al., 2021; Alambo et al., 2022; Krishna et al., 2020).

One of the central challenges in deploying these models in-the-wild is that rare words tend to have underestimated conditional probability during generation (Luong et al., 2014; Chintagunta et al., 2021; Holtzman et al., 2020). However, in high-stakes applications, many of these rare words are semantically important and need to be preserved. For example, some symptoms, diseases, and medications can be both rare and important (Mottaghi et al., 2020) (*e.g.* knowing that the patient is taking warfarin is extremely important, even if the word "warfarin" occurs infrequently).

Prior approaches for handling rare word generation utilize a copy mechanism (See et al., 2017; Joshi et al., 2020; Xu et al., 2020; Choi et al., 2021). This facilitates copying from the source text using a probabilistic switch to decide if the next output token is generated or copied from the input (See et al., 2017). However, it doesn't properly resolve the main challenge: not all rare tokens are important. Only specific rare tokens (*e.g.* warfarin) have a high probability of appearing in the reference sequence when found in the source sequence. In cases where the training data does not have enough structure to disambiguate which rare words are essential, the copy mechanism becomes overly extractive (Gehrmann et al., 2018; See et al., 2017).

Also relevant to this paper are previous works that integrate knowledge into language models (Duan et al., 2020; Liu et al., 2022). In entity-centric summarization, Keskar et al. (2019); Liu and Chen (2021) add key phrases to the prompt, which through the self-attention mechanism influence the output distribution. However, for prompts containing rare tokens, self-attention struggles to capture the prompt-reference dependency, and the marginal probability of rare tokens remains underestimated.

---

[1] Code is available as part of Supplemental Materials and will be available on Github after acceptance.

Joshi et al. (2020) extends this approach by not only explicitly including the medical concepts in the input sequence, but also adding a related term to the loss function. However, they still find that for rare tokens the model underestimates the conditional probability during generation.

Finally, dictionary look-up of rare and out-of-vocabulary words has been studied in Yu et al. (2022); Ruzzetti et al. (2022). However, these papers focus on finding good representations of specific tokens. In this paper, we tackle the problem of uplifting important rare tokens even when a good representation is not available.

We base our work on the premise that *specific* rare tokens (*e.g.* warfarin) have a high probability of appearing in the reference sequence if they also appear in the source sequence. The main questions we tackle in this paper are the following: *How do we know which rare tokens have a propensity to appear in both the source and the reference? How do we encode this information into the model?*

We study our approach in the healthcare setting, for the concrete problem of after-visit care instruction generation from a medical dialog between patient and medical professional. We define the medical concept utilization rate and utilization-rate-aware training objective in section 2, discuss the care plan generation problem and data collection in section 3, describe the sequence-to-sequence model setup in section 4, and report experimental results in section 5.

Our contributions are the following:

1. We are the first to explicitly focus on identifying and modeling specific rare tokens that appear in both the source and the reference. We call them “high utilization concepts.”
2. We propose a measure of “utilization rate” to identify tokens that comprise “high utilization concepts.” We use external knowledge to help with this computation as these tokens can be extremely rare.
3. We introduce a regularization term during training that leverages token utilization rate to uplift the conditional probability of important rare tokens.
4. We demonstrate the application of our approach to the concrete task of generating after-visit care instructions from medical professional-patient dialogue.

We observe performance improvement with both automatic metrics and human evaluation with medical experts.

2 APPROACH

In many sequence-to-sequence tasks, certain rare concepts have a high probability to appear in the reference sequence ($\mathbf{y}$) if they also appear in the source sequence ($\mathbf{x}$). We call these concepts “high utilization concepts” ($c \in C_{\text{HU}}$) and formally define them in Equation 1. These concepts are comprised of one or more tokens $c = [\nu_0, \nu_1, ...]$. We hypothesize that a source of factuality errors in many sequence-to-sequence tasks is that learned model underestimate the conditional probability of high utilization concepts $\hat{p}(y_i = \nu, |\mathbf{y}_{<i}, \mathbf{x}, \nu \in c, c \in \mathbf{x}, c \in C_{\text{HU}}) < p(...)$, where $\hat{p}$ denotes the model estimated probability and p is the true probability.

Definition 2.1 (High utilization concepts) *Given a universe of concepts $\mathcal{C}$, the set of high utilization concepts C_{HU} is defined as*

$$C_{\text{HU}} = \left\{ c \in \mathcal{C} : \frac{p(c \in \mathbf{y} | c \in \mathbf{x})}{p(c \in \mathbf{y})} \gg 1 \right\} \tag{1}$$

Equation 1 answers the question *“How do we know which rare tokens have a propensity to appear in both source and target?”* while at the same time it works for rare tokens.

This key insight leads us to define two goals for this work:

1. Develop a method for identifying high utilization concepts, C_{HU} for a dataset $\mathcal{D} = \{(\mathbf{x}^i, \mathbf{y}^i)\}_{i=1}^N$.
2. Develop a method for augmenting the training procedure of sequence-to-sequence models to correctly estimate the conditional probability of tokens forming high utilization concepts.

2.1 IDENTIFYING HIGH UTILIZATION CONCEPTS USING EXTERNALLY PROVIDED KNOWLEDGE

The major challenge in identifying high utilization concepts in real datasets is that the concepts we are interested in are present in very few examples. This means that it is hard to directly estimate $p(c \in \mathbf{y}|c \in \mathbf{x})$ and $p(c \in \mathbf{y})$ from Equation 1 due to the high variance. In particular, a frequency-based estimate of probability has an uncertainty proportional to $1/sqrt(N)$ where N is the number of samples for a given concept. However, these rare concepts can still be very impactful to the overall performance of the model. This is because, for a given reference, $\mathbf{y}$, it is unlikely that a *particular* high utilization concept will be present ($\forall c \in C_{\text{HU}}, p(c \in \mathbf{y}) \ll 1$), but it is also unlikely that *no* high utilization concept will be present ($\prod_{c \in C_{\text{HU}}} p(c \notin \mathbf{y}) \ll 1$). This is well documented in the medical domain, where medical concepts have a very long-tailed distribution (Prabhu et al., 2019; Mottaghi et al., 2020), yet may appear in almost every relevant sequence. As an illustration, imagine a list of medication instructions. Every instruction may have a different medication so no medication token appears more than once; however, each instruction is rendered useless if it doesn't include the relevant medication (*e.g.* see "Medication Plan" instructions in Figure 1).

To overcome this challenge, we propose computing what we call "utilization rate", r_ϕ, which we define in Equation 2. This function relies on the concept equivalence class map $\phi : C_{\text{sel}} \to \mathcal{E}$ where $C_{\text{sel}} \subseteq \mathcal{C}$ and $\mathcal{E}$ is a set of equivalence classes. (ϕ, C_{sel}, $\mathcal{E}$) cannot be derived from the data or the model, but instead are provided from an external source of knowledge. If ϕ is an identity (id) then $r_{\text{id}}(c_n) = \hat{p}(c_n \in \mathbf{y}|c_n \in \mathbf{x}), (\mathbf{x}, \mathbf{y}) \in \mathcal{D}$.

Definition 2.2 (Utilization rate) *The utilization rate of concept c_n is defined as*

$$r_\phi(c_n) = \frac{\sum_{c \in C_{\text{sel}}} \sum_{j=1}^{N} \mathbf{1}[c \in \mathbf{x}^j, c \in \mathbf{y}^j, \phi(c) = \phi(c_n)]}{\sum_{c \in C_{\text{sel}}} \sum_{j=1}^{N} \mathbf{1}[c \in \mathbf{x}^j, \phi(c) = \phi(c_n)]} \tag{2}$$

Here, Equation 2 tries to make the intuition from Equation 1 applicable to a real dataset. We generally cannot compute the lift because for rare words the dataset frequency derived probability estimates are poor.

Note that Equation 2 combines both externally provided knowledge (ϕ, C_{sel}, $\mathcal{E}$) and dataset derived values. This allows us to inject domain-specific information. Because concepts are mapped to equivalence classes, every concept in a particular equivalence class has the same utilization rate. If a concept $c_n \in C_{\text{sel}}$ has marginal probability to appear in the reference sequence that is much lower than $r_\phi(c_n)$ then it is a high utilization concept.

2.2 UTILIZATION-RATE-AWARE SEQ2SEQ TRAINING

Our analysis in section 5 (see Figure 3) shows that conventionally trained seq2seq models underestimate the utilization rate (r_ϕ) for many rare concepts. While we cannot optimize the utilization rate directly, we can optimize the approximate **marginal probability** $p(\nu|\mathbf{x})$ of a token ν given a source sequence $\mathbf{x}$, as seen in Equation 3.

$$p(\nu|\mathbf{x}) = \sum_{\mathbf{y}_{<t}} p(\nu|\mathbf{y}_{<t}) p(\mathbf{y}_{<t}) \approx \sum_{t=1}^{\|\mathbf{y}\|} p(\nu|\mathbf{y}_{<t}) p(\mathbf{y}_{<t}) \overset{p(\mathbf{y}_{<t}) \text{ is uniform}}{\approx} \frac{1}{\|\mathbf{y}\|} \sum_{t=1}^{\|\mathbf{y}\|} p(\nu|\mathbf{y}_{<t}) \tag{3}$$

Given the source sequence $\mathbf{x}$, the tokens for which we aim to optimize the marginal probability are $\{\nu \in c, c \in \mathbf{x} \cap C_{\text{HU}}\}$. We define the unweighted utilization loss.

Definition 2.3 (Unweighted utilization loss)

$$l_u(\mathbf{x}) = -\frac{1}{\|\{\nu \in c, c \in \mathbf{x} \cap C_{\text{HU}}\}\|} \sum_{\nu \in c, c \in (\mathbf{x} \cap C_{\text{HU}})} \log p(\nu|\mathbf{x}) \tag{4}$$

However, not all concepts in C_{HU} are equally likely to appear in the reference given their appearance in the source. To better reflect we also propose a weighted utilization loss where the weight for each token is determined by its utilization rate.

Definition 2.4 (Weighted utilization loss)

$$l_w(\mathbf{x}) = -\frac{\sum_{\nu \in c, c \in (\mathbf{x} \cap C_{\text{HU}})} r_\phi(c) \log p(\nu|\mathbf{x})}{\sum_{\nu \in c, c \in (\mathbf{x} \cap C_{\text{HU}})} r_\phi(c)} \tag{5}$$

Note that Equation 5 directly injects externally provided knowledge through its dependence on ϕ.

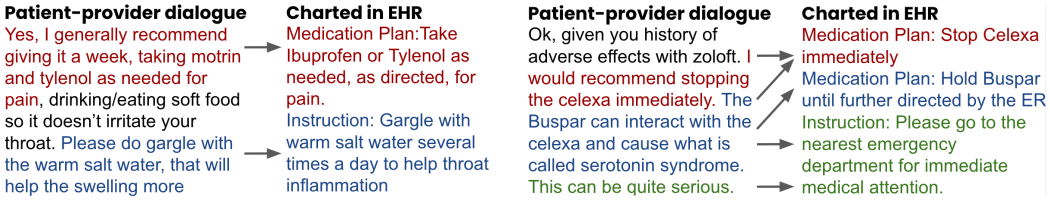
We use utilization loss as a regularization term and augment the objective function. We use $\alpha > 0$ to balance the strength of the regularization:

$$l(\mathbf{x}, \mathbf{y}) = l_{\text{nll}}(\mathbf{y}) + \alpha \cdot l_{u \text{ or } w}(\mathbf{x}) \tag{6}$$

where $l_{\text{nll}} = -\sum_{t=1}^{|\mathbf{y}|} \log p(y_t|\mathbf{y}_{<t}, \mathbf{x})$ and $l_{u \text{ or } w}$ is either l_u from Equation 4 or l_w from Equation 5.

3 AFTER-VISIT CARE INSTRUCTION GENERATION: TASK AND DATA DESCRIPTION

After-visit care instructions (care plan) are a set of actions (instructions) that a medical professional writes in the patient's electronic health record (EHR) as a follow-up to the patient's visit. A care plan often includes a list of medications with appropriate directions, further medical evaluations, or educational information for preventive care. Before writing the care plan, the medical professional discusses it with the patient, and together, they jointly agree on the next course of action. This joint decision-making implies that most of the necessary information for writing the care plan is already available in the conversation.



Yes, I generally recommend giving it a week, taking motrin and tylenol as needed for pain, drinking/eating soft food so it doesn't irritate your throat. Please do gargle with the warm salt water, that will help the swelling more

Medication Plan:Take Ibuprofen or Tylenol as needed, as directed, for pain.

Instruction: Gargle with warm salt water several times a day to help throat inflammation

Ok, given you history of adverse effects with zoloft. I would recommend stopping the celexa immediately. The Buspar can interact with the celexa and cause what is called serotonin syndrome. This can be quite serious.

Medication Plan: Stop Celexa immediately

Medication Plan: Hold Buspar until further directed by the ER

Instruction: Please go to the nearest emergency department for immediate medical attention.

(a) A relatively simple-to-chart example with each sentence corresponding to an instruction. Note synonym substitution of ibuprofen for motrin and the addition of timing to the gargling instruction.

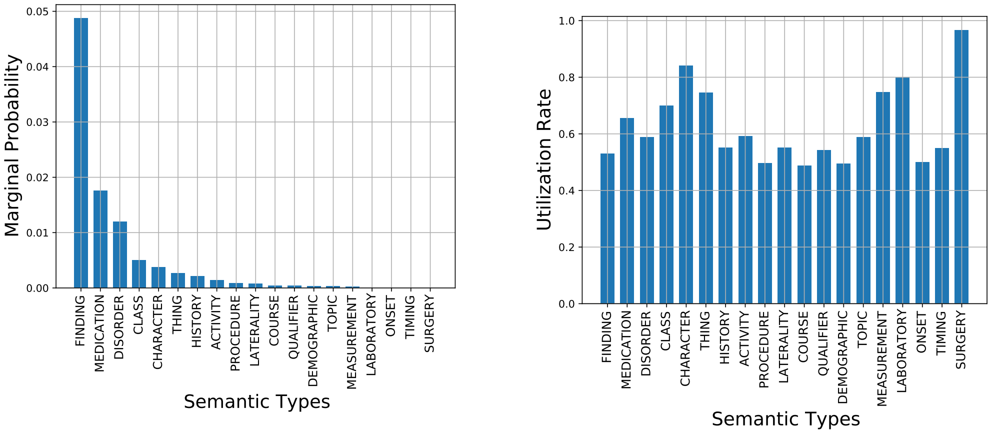
(b) A difficult-to-chart example with incomplete information and multiple dialogue sentences contributing to a single instruction.

Figure 1: Example conversation segments corresponding to care plan and corresponding instructions. Color represents the highest overlap between the sentence in the dialogue and the instruction. Arrows represent semantic relationship between the dialogue sentence and instruction. Note that these relationships between the dialog and the instructions are not available in the dataset.

In Figure 1, we show two examples. In each example, we present the (a) segment of the conversational dialog corresponding to provider messages discussing the care plan with the patient and (b) corresponding care plan charted in the EHR. We can see that the instructions are written in a directive format, using action verbs and often paraphrasings of the corresponding text in the dialogue. The care plan does not always have all the medical concepts mentioned in the conversation. In the first example, "serotonin syndrome" and "Celexa" are rare, but the care plan includes only the latter. We need a model that is robust to rare medical concepts and can discern which knowledge needs to be carried forward.

We tackle the problem of taking the relevant section in the conversations corresponding to the care plan as input and automatically derive care plan instructions that the medical professionals can approve. We do not assume access to 1-1 mappings between the sentences in the conversation to the care plan instructions. However, we develop a method to derive a dataset of 1-1 mappings, albeit noisy, which we use for model training.

Dataset construction. We use a dataset with 14K medical professional-patient encounters collected on a virtual primary care platform. Each encounter has a text-based conversation between the medical



(a) Average marginal probability of every semantic type.

(b) Semantic type utilization rates.

Figure 2: Empirical concept marginal probabilities and utilization rates estimated from the dataset.

professional and the patient. We applied an in-house conversation discourse parser to extract only those dialogue turns from the medical professional's corresponding to the care plan discussion [2]. We also have the associated care plans written from the patient's electronic health record for that encounter. On average, each encounter has 9 dialogue turns corresponding to care plans and 4 care plan instructions.

We need a parallel corpus with pairs of dialogue turns and care plan instructions for our model. Getting manual annotations for each encounter would be expensive as it requires expert knowledge. Therefore, we automatically construct a paired dataset, albeit noisily, from the paired encounter level care plan and provider dialog turns. We get sentence-level embeddings for every sentence in each turn and instructions in the care plan and pair those with the highest cosine similarity (We provide additional details in the Supplementary Material). At the end of this, we have 48,000 source-reference pairs, where the source is a sentence in the conversational dialog and reference is the mapped instruction. We randomly sample 3000 pairs for testing, 1000 for validation, and the remaining 44,000 pairs for training.

We use medical concepts from UMLS Bodenreider (2004) and in particular SNOMED-CT and RXNorm ontologies. The synonyms are pooled from all ontologies in UMLS that map to the corresponding concept in SNOMED-CT and RXNorm.

To identify the concepts, we use an in-house lookup-based concept recognizer. It uses a sliding window strategy to find maximal matches of text corresponding to medical concepts and their synonyms. It ignores stop words while doing the match. Finally, it has an agglomeration step that leverages a concept hierarchy. If we have overlapping spans corresponding to two concepts where one is a child of another (eg "lower abdominal pain" and "abdominal pain") then only the more specific concept is extracted. If two different concepts have a span overlap and are not hierarchically related, then the concept linking is greedily selected with the concept on the left being given priority.

Identifying high utilization concepts. We limit C_{sel} to only medical concepts and choose ϕ such that it maps them to their SNOMED CT semantic types (which informs our choice of $\mathcal{E}$). In our case study this narrows down 758 unique medical concepts to their 19 semantic types. The marginal probability $p(c \in \mathbf{y})$ for each semantic type c is shown in Figure 2a while the utilization rates are shown in Figure 2b. Comparing them we can see that utilization rates are 10-100x larger than the marginal probabilities. This suggests that all medical concepts are part of high utilization tokens set ($C_{\text{HU}} = C_{\text{sel}}$). It also means that many kinds of medical concepts that are present in the source sequence do not get generated in the output sequence, which drastically hurts medical correctness.

---

[2] A manuscript describing this parser is currently submitted for publication and a reference will be added to the camera-ready version of this paper.

4 EXPERIMENTAL SETUP

We follow the standard practice (Ott et al., 2018) of training our sequence-to-sequence models using FairSeq framework (Ott et al., 2019). We use byte-pair encoding implemented in the fastBPE package (Sennrich et al., 2016). We use a transformer architecture for our model and train models on our data from scratch[3].

Model architecture We use the `transformer_iwslt_de_en` architecture in FairSeq for experiments. It consists of 6 encoder and decoder layers with 4 self-attention heads followed by feed-forward transformations. Both encoder and decoder use embeddings of size 512 while the input and output embeddings are not shared. Both the encoder and decoder use learned positional embedding. We early-stop training based on the validation performance. Evaluation is done on the test set.

Training We use Adam optimizer (Kingma and Ba, 2015) with $\beta_1 = 0.9$ and $\beta_2 = 0.98$. We use the inverse square root learning scheduler with 4,000 warm-up steps. We use the initial learning rate of 5×10^{-4}, dropout rate of 0.3 (Srivastava et al., 2014) , and weight decay with its rate set to 10^{-4}. We use label smoothing with 0.1 of probability smoothed uniformly during training. We modify the training objective Equation 6 by adding oversmoothing loss (Kulikov et al., 2021) with a coefficient of 0.9 and unlikelihood loss (Welleck et al., 2019) with a coefficient of 0.5. All training was performed on VMs with single V100 GPUs, we estimate 200 GPU hours as the total amount required for the completion of this work.

Early stopping We use early stopping for model selection based on the value of the objective function computed on the validation set. We evaluate the model on the development set every 2K updates ($\sim$4K tokens per update). We stop training when the objective has not improved over more than 5 consecutive validation runs. It takes approximately 75K updates to an early stop.

Decoding We use beam search implementation from FairSeq. We decode using the beam size of 5. We set the lower- and upper-bound of a generated output to be, respectively, 0 and $1.2 \cdot ||\mathbf{x}|| + 10$. We do not use either length normalization or length penalty since we apply oversmoothing loss.

Lexically constrained decoding baseline Apart from using the unregularized version of the model as a baseline, we compare the proposed approach with the lexically constrained decoding approach (Post and Vilar, 2018). We stick to the `LexicallyConstrainedBeamSearch` implementation of the Dynamic Beam Allocation (DBA) algorithm that ensures the presence of provided tokens in the generated output. DBA implements an optimized version of the Grid Beam Search (Hokamp and Liu, 2017). DBA is training-agnostic and is used only during generation. We apply DBA for the baseline model. Given the non-uniform distribution of utilization rates, for each source we leave only medical concepts c with $r_{\text{id}}(c) > \tau$ for some threshold τ. We report results for $\tau = 0.6$, which we select by running an extensive grid search.

5 RESULTS

5.1 EFFECT OF KNOWLEDGE INJECTION DURING TRAINING ON MODEL'S UTILIZATION RATE

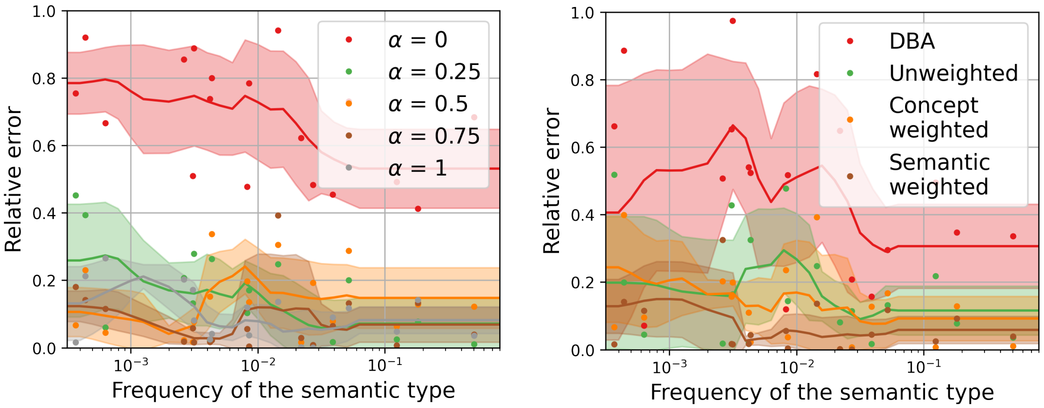
We evaluate whether the knowledge injection through regularization (subsection 2.2) has the desired effect of improving model estimate of the utilization rate, r_ϕ. Because the test set is too small to effectively estimate per-concept utilization rate, we instead compute it for semantic types. In Figure 3 we use semantic relative error (Equation 7) to compare models trained with $\alpha \in \{0, 0.25, 0.5, 0.75, 1\}$ that either use unweighted loss l_u (which uplifts all medical concepts equally, "Unweighted") or a weighted loss l_w with the ϕ being identity ("Concept weighted") or mapping concepts to semantic types ("Semantic weighted"). In addition, as a baseline we also compare an unregularized model that uses DBA for generation ("DBA"). For a detailed breakdown of relative errors for each combination see the Supplementary Material.

---

[3] Informally, we also tried a pre-trained BART (Lewis et al., 2019) but the results were worse.

Definition 5.1 (Semantic relative error) *Relative error for semantic type s computed from $\hat{r}_\phi$ estimated from model derived output sequences and r_ϕ estimated from reference sequences. c_s is any concept for which $\phi(c) = s$ holds and the value of ϵ_s in not dependent on the choice of c_s.*

$$\epsilon_s = \frac{\|\hat{r}_\phi(c_s) - r_\phi(c_s)\|}{r_\phi(c_s)} \tag{7}$$



(a) Relative error in the utilization rate for each regularizer strength α. Note that $\alpha = 0$ means there is no regularization.

(b) Relative error in the utilization rate for each model (and best α for that model).

Figure 3: Relative errors in the utilization rates for different semantic types plotted as a function of the frequency of the semantic type. The trend-line and uncertainty are computed with a linearly interpolated moving average window.

In Figure 3a we present the relative error for different α as a function of semantic type frequency in the test set. For each point (a given semantic type and α) we take the lowest relative error among {“Unweighted”, “Concept weighted”, and “Semantic weighted”}. The highest relative errors are seen for $\alpha = 0$, which corresponds to no regularization. For other values of α the difference is not statistically significant, although, for very rare semantic types, $\alpha = 0.25$ appears to perform worse than models with higher regularization strength. This shows that our external knowledge informed regularization has a significant impact on a relative error, but the utilization rate estimate is not sensitive to the exact weight of the regularization term.

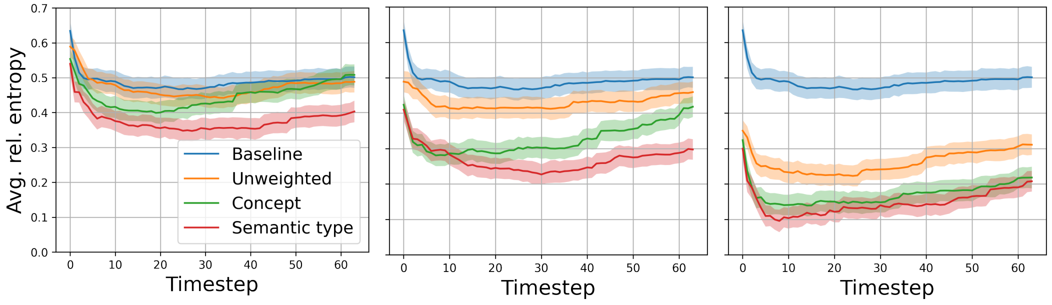
In Figure 3b we present relative error for different training procedures, {“Unweighted”, “Concept weighted”, and “Semantic weighted”}, as well as a baseline of “DBA.” For each point (a given semantic type and training procedure) we choose an α that gives the lowest relative error. We find that “DBA" baseline, which is a constrained generation procedure applied to an unregularized model, performs worse than any of the regularized models, although it does outperform the unregularized model ($\alpha = 0$ in Figure 3a). While not significant, we also see that for rare semantic types “Semantic weighted” seems to perform the best, which aligns with our expectation that the utilization rate is hard to estimate for very rare concepts.

5.2 EFFECT OF KNOWLEDGE INJECTION DURING TRAINING ON MODEL’S UNCERTAINTY

We analyze the effect of utilization regularization on the model’s uncertainty at every timestep. Uncertainty at timestep t is defined as an entropy of model’s distribution on each timestep t (here $\mathbf{y}_{<t}$ is the decoded sequence up to t-th timestep, y is an arbitrary token from the target vocabulary):

$$H_t(\mathbf{y}_{<t}, \mathbf{x}) = -\sum_{y} p(y|\mathbf{y}_{<t}, \mathbf{x}) \log p(y|\mathbf{y}_{<t}, \mathbf{x}) \tag{8}$$

We consider the defined uncertainty on earlier timesteps, where the model’s distribution is closer to marginal. As the proposed method pushes up the marginal probability of the medical concepts, we claim that models’ uncertainty decreases with the regularization. Moreover, care plan instructions



(a) $\alpha = 0.25$ (b) $\alpha = 0.5$ (c) $\alpha = 1.0$

Figure 4: Entropy of the conditional distribution $p(y|\mathbf{y}_{<t}, \mathbf{x})$ with respect to different α values. Filled regions denote the standard deviation across training runs according to section 4.

typically introduce crucial concepts at the beginning of an instruction. Thus, we claim that early timesteps uncertainty matters for the precise decoding of instructions.

This is confirmed by Figure 4. We observe that uncertainty drops monotonically as the α weight increases. In particular, uncertainty on early timesteps heavily drops as a result of utilization minimization. Hence, the model becomes more confident in selecting principal concepts at the beginning of an instruction. In contrast to the baseline, all regularized models' uncertainty start to increase for $t > 10$. As fewer concepts appear in the instruction end, the marginal probability maximization flattens the conditional distribution. However, the uncertainty does not degrade in comparison to the baseline. Thus, the proposed regularization effectively improves the confidence of the model on early timesteps.

5.3 RESULTS ON CARE PLAN INSTRUCTIONS TASK

	α	BERTScore	Concept-F1	GPT-2 Perplexity
Baseline	0.0	22.48 ±0.66	57.43±3.73	**5.53**±0.04
DBA	-	23.59±0.28	**79.83**±0.43	11.96±0.05
Unweighted (ours)	0.25	25.09±0.69	58.19±2.11	5.91±0.07
	0.5	25.42±0.56	58.91±6.83	5.65±0.03
	0.75	26.22±0.35	60.83±5.96	6.28±0.02
	1.0	26.74±0.43	61.05±7.48	6.18±0.05
Concept weighted (ours)	0.25	28.29±0.19	60.87±3.86	6.93±0.05
	0.5	28.19±0.20	60.36±2.03	8.49±0.05
	0.75	28.08±0.15	64.09±1.85	7.95±0.080
	1.0	27.82±0.25	63.05±2.49	9.37±0.10
Semantic weighted (ours)	0.25	28.97±0.56	69.10±2.12	7.01±0.29
	0.5	**30.54**±0.78	74.98±3.91	6.84±0.03
	0.75	**31.48**±0.86	75.77±3.30	6.96±0.11
	1.0	**30.59**±0.63	75.02±2.18	6.94±0.12

Table 1: Automated metrics scores for different model setups. We report average score and standard deviation over five random seeds. We highlight in bold the best average and all scores having overlapped standard deviation intervals with the best score.

Automated evaluation: The precise and complete concepts utilization directly affects the quality of instruction. We first quantify the quality by calculating automatic metrics to judge the relevance, fluency, and concept utilization rate in comparison to the reference instructions. We use BERTScore (Zhang et al., 2019) to estimate the similarity between reference and candidate, GPT-2 perplexity for (Nguyen, 2021) to assess the coherence (fluency) of the candidate, and concept overlap (Joshi et al., 2020) to measure the percentage of medical concepts used in both candidate in reference.

Table 1 presents the automatic evaluation results. The scores indicate that incorporating knowledge correlates with relevance and concept overlap. We highlight three observations. First, the regularization is effective in terms of quality and concept overlap. We observe significant quality improvement compared to both the baseline and DBA. Moreover, weighted versions of the model outperform the unweighted setup. Thus, injecting more knowledge into the model, such as empirical utilization weights, results in better quality. Second, the impact of the regularization hardly depends on the α weight. Third, the GPT-2 perplexity degrades. This demonstrates that the regularization impacts the model distribution, so the fluency of the model may deteriorate. This trade-off, however, has no negative impact on the quality given the improved BERTScore. For qualitative results, please see the Supplementary Material.

Medical experts evaluation: To get a more precise medical assessment, we conduct human evaluation with medical experts. We randomly sample 100 dialogues from the test set and generate candidates with each model setup setting $\alpha = 1.0$. We ask five doctors to evaluate the relevance to the dialogue, medical usability (if the generated instruction can be used in any care plan), and grammatical correctness (fluency) on a scale from 1 to 5. Additionally, we ask assessors to indicate degenerate generations, i.e., premature or repetitive sequences. Exact questions and interface screenshots can be found in the Supplementary Material.

As shown in Table 2, we claim that both weighted versions achieve significant improvement in relevance and usability, which are target medical metrics. In contrast to the GPT-2 perplexity, medical experts report equal fluency for all models but DBA. We explain this discrepancy with vocabulary shift as GPT-2 is not trained on a healthcare corpus. Finally, utilization rate regularization does not affect the number of degenerate outputs. Hence, the proposed solution effectively induces knowledge in the model distribution without corrupting generated text correctness. This is not true for DBA, which struggles from a lack of coherence and degenerate outputs while producing more relevant and usable instructions.

	Relevance	Usability	Fluency	Degeneracies, %
Baseline	2.50±0.12	3.18±0.27	**4.17**±0.14	**0.10**±0.01
DBA	3.36±0.15	3.35±0.16	3.91±0.18	0.21±0.05
Unweighted (ours)	3.56±0.12	3.21±0.28	**4.26**±0.08	**0.10**±0.02
Concept weighted (ours)	**3.79**±0.06	3.72±0.05	**4.37**±0.16	**0.12**±0.02
Semantic weighted (ours)	**3.78**±0.14	**3.99**±0.19	**4.42**±0.13	**0.12**±0.012

Table 2: Evaluation using medical experts. Fluency, Usability, and Relevance are scored on a scale from 1 to 5. We also report the percentage of premature or repetitive outputs (Degeneracies). We report average score and standard deviation of experts' scores. We highlight in bold the best average and all scores having overlapped standard deviation intervals with the best score.

6 CONCLUSION

In this work, we tackle the problem of under-generation of rare but important tokens in sequence-to-sequence models. We show that external knowledge can be effectively injected into the sequence-to-sequence models and mitigate the problem of lexical precision. We characterize the problem by identifying a set of low-frequency but important concepts and defining their utilization rate, which estimates the probability of a concept that is present in the source to be also present in the reference. We confirm that modern well-trained sequence-to-sequence models suffer from under-estimating utilization rates, and propose a way to directly maximize it during training. We design a differentiable proxy based on the marginal entropy and propose a regularized training objective. Since some concepts may be omitted from the reference, we extend the approach by applying weights, which restrict the regularization impact of low-utilized concepts or their semantic types.

We perform a case study in automatic care plan generation from medical dialogues. We experiment with a custom internal dataset and observe the effectiveness of the approach. We also compare a previous approach for external knowledge injection – dynamic beam allocation (DBA). First, we find that regularization improves the model's utilization rate by pushing it closer to the empirical values observed in reference sequences. Second, regularization reduces the model's uncertainty at early timesteps: exactly where concepts are typically introduced. Third, we observed a significant (in terms of standard deviations) quality improvement. More specifically, we did a human evaluation of relevance, concept overlap, medical usability, and fluency using five medical experts. The results revealed the enhanced relevance and usability of generated instructions while, unlike DBA, maintaining high fluency and low degeneracy.

Ethics Statement: This work was done as part of a quality improvement activity as defined in 45CFR §46.104(d)(4)(iii) – "health care operations" secondary research.

Reproducibility statement: Code used for training regularized sequence-to-sequence models in this paper is available as part of Supplemental Materials and will be available on Github after acceptance. However, data will not be shared due to patient privacy and HIPAA compliance. as it contains significant amount of Patient Health Information (PHI) and cannot be shared.

REFERENCES

Amanuel Alambo, Tanvi Banerjee, Krishnaprasad Thirunarayan, and Mia Cajita. 2022. Improving the factual accuracy of abstractive clinical text summarization using multi-objective optimization.

Kristen Bell, Jenny Hong, Nick McKeown, and Catalin Voss. 2021. The recon approach: A new direction for machine learning in criminal law. In *Berkeley Technology Law Journal*.

Olivier Bodenreider. 2004. The Unified Medical Language System (UMLS): Integrating biomedical terminology. *Nucleic Acids Research*, 32.

Jai Chintagunta, Namit Katariya, Xavier Amatriain, and Anitha Kannan. 2021. Medically aware gpt-3 as a data generator for medical dialogue summarization. *Machine Learning for Healthcare*.

Sanghyuk Choi, Jeong-In Hwang, Hyungjong Noh, and Yeonsoo Lee. 2021. May the force be with your copy mechanism: Enhanced supervised-copy method for natural language generation. *CoRR*, abs/2112.10360.

Rhys Compton, Ilya Valmianski, Li Deng, Costa Huang, Namit Katariya, Xavier Amatriain, and Anitha Kannan. 2021. Medcod: A medically-accurate, emotive, diverse, and controllable dialog system. In *Proceedings of Machine Learning for Health*, volume 158 of *Proceedings of Machine Learning Research*, pages 110–129. PMLR.

Chenhe Dong, Yinghui Li, Haifan Gong, Miaoxin Chen, Junxin Li, Ying Shen, and Min Yang. 2021. A survey of natural language generation. *CoRR*, abs/2112.11739.

Yu Duan, Canwen Xu, Jiaxin Pei, Jialong Han, and Chenliang Li. 2020. Pre-train and plug-in: Flexible conditional text generation with variational auto-encoders. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 253–262, Online. Association for Computational Linguistics.

Erkut Erdem, Menekse Kuyu, Semih Yagcioglu, Anette Frank, Letitia Parcalabescu, Barbara Plank, Andrii Babii, Oleksii Turuta, Aykut Erdem, Iacer Calixto, Elena Lloret, Elena-Simona Apostol, Ciprian-Octavian Truică, Branislava Šandrih, Sanda Martinčić-Ipšić, Gábor Berend, Albert Gatt, and Grăzina Korvel. 2022. Neural natural language generation: A survey on multilinguality, multimodality, controllability and learning. *J. Artif. Int. Res.*, 73.

Sebastian Gehrmann, Yuntian Deng, and Alexander Rush. 2018. Bottom-up abstractive summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, Brussels, Belgium. Association for Computational Linguistics.

Chris Hokamp and Qun Liu. 2017. Lexically constrained decoding for sequence generation using grid beam search. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1535–1546, Vancouver, Canada. Association for Computational Linguistics.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *International Conference on Learning Representations*.

Anirudh Joshi, Namit Katariya, Xavier Amatriain, and Anitha Kannan. 2020. Dr. summarize: Global summarization of medical dialogue by exploiting local structures. *EMNLP-Findings*.

Nitish Shirish Keskar, Bryan McCann, Lav R. Varshney, Caiming Xiong, and Richard Socher. 2019. Ctrl: A conditional transformer language model for controllable generation.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR*.

Kundan Krishna, Sopan Khosla, Jeffrey P. Bigham, and Zachary C. Lipton. 2020. Generating soap notes from doctor-patient conversations.

Ilia Kulikov, Maksim Eremeev, and Kyunghyun Cho. 2021. Characterizing and addressing the issue of oversmoothing in neural autoregressive sequence modeling.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Xiaochen Liu, Yu Bai, Jiawei Li, Yinan Hu, and Yang Gao. 2022. Psp: Pre-trained soft prompts for few-shot abstractive summarization.

Zhengyuan Liu and Nancy Chen. 2021. Controllable neural dialogue summarization with personal named entity planning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 92–106, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Thang Luong, Ilya Sutskever, Quoc V. Le, Oriol Vinyals, and Wojciech Zaremba. 2014. Addressing the rare word problem in neural machine translation. *CoRR*, abs/1410.8206.

Ali Mottaghi, Prathusha K. Sarma, Xavier Amatriain, Serena Yeung, and Anitha Kannan. 2020. Medical symptom recognition from patient text: An active learning approach for long-tailed multilabel distributions. *CoRR*, abs/2011.06874.

An Nguyen. 2021. Language model evaluation in open-ended text generation.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*.

Myle Ott, Sergey Edunov, David Grangier, and Michael Auli. 2018. Scaling neural machine translation. *Proceedings of the Third Conference on Machine Translation: Research Papers*.

Matt Post and David Vilar. 2018. Fast lexically constrained decoding with dynamic beam allocation for neural machine translation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1314–1324, New Orleans, Louisiana. Association for Computational Linguistics.

Viraj Prabhu, Anitha Kannan, Geoffrey J. Tso, Namit Katariya, Manish Chablani, David A. Sontag, and Xavier Amatriain. 2019. Open set medical diagnosis. *CoRR*, abs/1910.02830.

Elena Sofia Ruzzetti, Leonardo Ranaldi, Michele Mastromattei, Francesca Fallucchi, Noemi Scarpato, and Fabio Massimo Zanzotto. 2022. Lacking the embedding of a word? look it up into a traditional dictionary. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2651–2662, Dublin, Ireland. Association for Computational Linguistics.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Jia Tracy Shen, Michiharu Yamashita, Ethan Prihar, Neil T. Heffernan, Xintao Wu, and Dongwon Lee. 2021. Mathbert: A pre-trained language model for general NLP tasks in mathematics education. *CoRR*, abs/2106.07340.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958.

Ilya Valmianski, Nave Frost, Navdeep Sood, Yang Wang, Baodong Liu, James J. Zhu, Sunil Karumuri, Ian M. Finn, and Daniel S. Zisook. 2021. Smarttriage: A system for personalized patient data capture, documentation generation, and decision support. In *Proceedings of Machine Learning for Health*, volume 158 of *Proceedings of Machine Learning Research*, pages 75–96. PMLR.

Sean Welleck, Ilia Kulikov, Stephen Roller, Emily Dinan, Kyunghyun Cho, and Jason Weston. 2019. Neural text generation with unlikelihood training.

Liqiang Xiao, Jun Ma, Xin Luna Dong, Pascual Martínez-Gómez, Nasser Zalmout, Wei Chen, Tong Zhao, Hao He, and Yaohui Jin. 2021. End-to-end conversational search for online shopping with utterance transfer. *CoRR*, abs/2109.05460.

Song Xu, Haoran Li, Peng Yuan, Youzheng Wu, Xiaodong He, and Bowen Zhou. 2020. Self-attention guided copy mechanism for abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1355–1362, Online. Association for Computational Linguistics.

Wenhao Yu, Chenguang Zhu, Yuwei Fang, Donghan Yu, Shuohang Wang, Yichong Xu, Michael Zeng, and Meng Jiang. 2022. Dict-BERT: Enhancing language model pre-training with dictionary. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1907–1918, Dublin, Ireland. Association for Computational Linguistics.

Denghui Zhang, Zixuan Yuan, Yanchi Liu, Fuzhen Zhuang, Haifeng Chen, and Hui Xiong. 2020. E-bert: A phrase and product knowledge enhanced language model for e-commerce.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert.