Zero-Shot Clinical Acronym Expansion via Latent Meaning Cells

Griffin Adams griffin.adams@columbia.edu
Mert Ketenci mert.ketenci@columbia.edu
Shreyas Bhave sab2323@cumc.columbia.edu
Adler Perotte adler.perotte@columbia.edu
Noémie Elhadad noemie.elhadad@columbia.edu
Columbia University, New York, NY, US

Editors: Emily Alsentzer$^{\otimes}$, Matthew B. A. McDermott$^{\otimes}$, Fabian Falck, Suproteem K. Sarkar, Subhrajit Roy‡, Stephanie L. Hyland‡

Abstract

We introduce Latent Meaning Cells, a deep latent variable model which learns contextualized representations of words by combining local lexical context and metadata. Metadata can refer to granular context, such as section type, or to more global context, such as unique document ids. Reliance on metadata for contextualized representation learning is apropos in the clinical domain where text is semi-structured and expresses high variation in topics. We evaluate the LMC model on the task of zero-shot clinical acronym expansion across three datasets. The LMC significantly outperforms a diverse set of baselines at a fraction of the pre-training cost and learns clinically coherent representations. We demonstrate that not only is metadata itself very helpful for the task, but that the LMC inference algorithm provides an additional large benefit.

Keywords: variational inference, clinical acronyms, representation learning

1. Introduction

Pre-trained language models have yielded remarkable advances in multiple natural language processing (NLP) tasks. Probabilistic models such as LDA (Blei et al., 2003), on the other hand, can uncover latent document-level topics. In topic models, words are drawn from shared topic distributions at the document level, whereas in language models, word semantics arise from co-occurrence with other words in a tighter window.

We build upon both approaches and introduce Latent Meaning Cells (LMC), a deep latent variable model which learns a contextualized representation of a word by combining evidence from *local context* (i.e., the word and its surrounding words) and *document-level metadata*. We use the term metadata to generalize the framework because it may vary depending on the domain and application. Metadata can refer to a document itself, as in topic modeling, document categories (i.e, articles tagged under *Sports*), or structures within documents (i.e., section headers). Incorporating latent factors into language modeling allows for direct modeling of the inherent uncertainty of words. As such, we define a latent meaning cell as a Gaussian embedding jointly drawn from word and metadata prior densities. Conditioned on a central word and its metadata, the latent meaning cell identifies surrounding words as in a generative Skip-Gram model (Mikolov et al., 2013a; Bražinskas et al., 2018). We approximate posterior densities by devising an amortized variational distribution over the latent meaning cells. The approximate posterior can best be viewed as the embedded word sense based on local context and metadata. In this way, the LMC is non-parametric in the number of latent meanings per word type.

We motivate and develop the LMC model for the task of zero-shot clinical acronym expansion. Formally, we consider the following task: given clinical text containing an acronym, select the acronym's most likely expansion from a predefined expansion set. It is analogous to word sense disambiguation, where sense sets are provided by a medical acronym expansion inventory. This task is important because clinicians frequently use acronyms with diverse meanings across contexts, which makes robust text processing difficult (Meystre et al., 2008; Demner-Fushman and Elhadad, 2016). Yet clinical texts are highly structured, with established section headers across note types and hospitals (Weed, 1968). Section headers can serve as a helpful clue in uncovering latent acronym expansions. For instance, the abbreviation *Ca* is more likely to stand for *calcium* in a *Medications* section whereas it may refer to *cancer* under the *Past Medical History* section. Prior work has supplemented local word context with document-level features: latent topics (Li et al., 2019) and bag of words (Skreta et al., 2019), rather than section headers.

In our experiments, we directly assess the importance of section headers on zero-shot clinical acronym expansion. Treating section headers as metadata, we pre-train the LMC model on MIMIC-III clinical notes with extracted sections. Using three test sets, we compare its ability to uncover latent acronym senses to several baselines pre-trained on the same data. Since labeled data is hard to come by, and clinical acronyms evolve and contain many rare forms (Skreta et al., 2019; Townsend, 2013), we focus on the zero-shot scenario: evaluating a model's ability to align the meaning of an acronym in context to the unconditional meaning of its target expansion. No models are fine-tuned on the task. We find that metadata complements local word-level context to improve zero-shot performance. Also, metadata and the LMC model are synergistic - the model's success is a combination of a helpful feature (section headers) and a novel inference procedure.

We summarize our primary contributions: **(1)** We devise a contextualized language model which jointly reasons over words and metadata. Previous work has learned document-level representations. In contrast, we explicitly condition the meaning of a word on these representations. **(2)** Defining metadata as section headers, we evaluate our model on zero-shot clinical acronym expansion and demonstrate superior classification performance. With relatively few parameters and rapid convergence, the LMC model offers an efficient alternative to more computational intensive models on the task. **(3)** We publish all code[1] to train, evaluate, and create test data, including regex-based toolkits for reverse substitution and section extraction. This study and use of materials was approved by our institution's IRB.

2. Related Work

Word Embeddings. Pre-trained language models learn contextual embeddings through masked, or next, word prediction (Peters et al., 2018a; Devlin et al., 2019; Yang et al., 2019; Bowman et al., 2019; Liu et al., 2019; Radford et al., 2019). Recently, SenseBert (Levine et al., 2019) leverages WordNet (Miller, 1998) to add a masked-word sense prediction task as an auxiliary task in BERT pre-training. While these models represent words as point embeddings, Bayesian language models treat

[1] https://github.com/griff4692/LMC

embeddings as distributions. Word2Gauss defines a normal distribution over words to enable the representation of words as soft regions (Vilnis and McCallum, 2014). Other works directly model polysemy by treating word embeddings as mixtures of Gaussians (Tian et al., 2014; Athiwaratkun and Wilson, 2017; Athiwaratkun et al., 2018). Mixture components correspond to the different word senses. But most of these approaches require setting a fixed number of senses for each word. Non-parametric Bayesian models enable a variable number of senses per word (Neelakantan et al., 2014; Bartunov et al., 2016). The Multi-Sense Skip Gram model (MSSG) creates new word senses online, while the Adaptive Skip-Gram model (Bartunov et al., 2016) uses Dirichlet processes. The Bayesian Skip-gram Model (BSG) proposes an alternative to modeling words as a mixture of discrete senses (Bražinskas et al., 2018). Instead, the BSG draws latent meaning vectors from center words, which are then used to identify context words.

Embedding models that incorporate global context have also been proposed (Le and Mikolov, 2014; Srivastava et al., 2013; Larochelle and Lauly, 2012). The generative models Gaussian LDA, TopicVec, and the Embedded Topic Model (ETM) integrate embeddings into topic models (Blei et al., 2003). ETM represents words as categorical distributions with a natural parameter equal to the inner product between word and assigned topic embeddings (Dieng et al., 2019); Gaussian LDA replaces LDA's categorical topic assumption with multivariate Gaussians (Das et al., 2015); TopicVec can be viewed as a hybrid of LDA and PSDVec (Li et al., 2016). While these models make inference regarding the latent topics of a document given words, the LMC model makes inference on meaning given both a word and metadata.

Clinical Acronym Expansion. Acronym expansion—mapping a Short Form (SF) to its most likely Long Form (LF)— is a task within the problem of word-sense disambiguation (Camacho-Collados and Pilehvar, 2018). For instance, the acronym PT refers to "patient" in *"PT is 80-year old male,"* whereas it refers to "physical therapy" in *"prescribed PT for back pain."* Traditional supervised approaches to clinical acronym expansion consider only the local context (Joshi et al., 2006). Li et al. (2019) leverage contextualized ELMo, with attention over topic embeddings, to achieve strong performance after fine-tuning on a randomly sampled MIMIC dataset. On the related task of biomedical entity linking, the LATTE model (Zhu et al., 2020) uses an ELMo-like model to map text to standardized entities in the UMLS metathesaurus (Bodenreider, 2004). Skreta et al. (2019) create a reverse substitution dataset and address class imbalances by sampling additional examples from related UMLS terms. Jin et al. (2019b) fine-tune bi-ELMO (Jin et al., 2019a) with abbreviation-specific classifiers on Pubmed abstracts.

3. Latent Meaning Cells

As shown in Figure 1, latent meaning cells postulate both words and metadata as mixtures of latent meanings.

3.1. Motivation

In domains where text is semi-structured and expresses high variation in topics, there is an opportunity to consider context between low-level lexical and global document-level. Clinical texts from the electronic health record represent a prime example. Metadata, such as section header and note type, can offer vital clues for polysemous words like acronyms. Consequently, we posit that a word's latent meaning directly depends on its metadata.

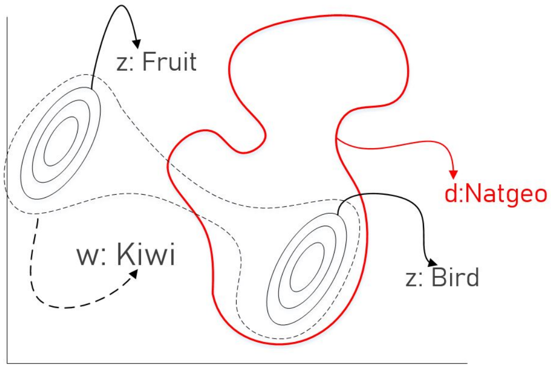


Figure 1: The word "kiwi" can take on multiple meanings. When used inside a National Geographic article, its latent meaning is restricted to lie inside the red distribution and is closer to "bird" than "fruit".

We define a latent meaning cell (lmc)[2] as a latent Gaussian embedding jointly drawn from word and metadata prior densities. The lmc represents a draw of an embedded word sense based on metadata. In a Skip-Gram formulation, we assume that context words are generated from the lmc formed by the center word and corresponding metadata. Context words, then, are conditionally independent of center words and metadata given the lmc.

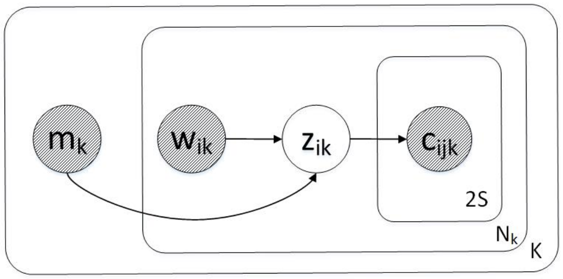
3.2. Notation

A word is the atomic unit of discrete data and represents an item from a fixed vocabulary. A word is denoted as w when representing a center word, and c for a context word. $\boldsymbol{c}$ represents the set of context words relative to a center word w. In different contexts, each word operates as both a center word and a context word. For our purposes, metadata are pseudo-documents which contain a sequence of N words denoted by $m = (w_1, w_2, ..., w_N)$ where w_n is the n^{th} word. (A.2 visually depicts metadata). A corpus is a collection of K metadata denoted by $D = \{m_1, m_2, ..., m_K\}$.

3.3. Latent Variable Setup

We rely on graphical model notation as a convenient tool for describing the specification of the objective, as is commonly done in latent variable model work (e.g., (Bražinskas et al., 2018)). Using the notation from Section 3.2, we illustrate the pseudo-generative[3] process in plate notation and story form.

Figure 2: LMC Plate Notation.



Algorithm 1 Pseudo-Generative Story

```
for $k = 1...K$ do
    Draw metadata $m_k \sim \text{Cat}(\gamma)$
    for $i = 1...N_k$ do
        Draw word $w_{ik} \sim \text{Cat}(\alpha)$
        Draw lmc $z_{ik} \sim p(z_{ik}|w_{ik}, m_k)$
        for $j = 1...2S$ do
            Draw context word $c_{ijk} \sim p(c_{ijk}|z_{ik})$
```

S is the window size from which left-right context words are drawn. The factored joint distribution between observed and unobserved random variables $P(M, W, C, Z)$ is:

$$\prod_{k=1}^{K} p(m_k) \prod_{i=1}^{N_k} p(w_{ik}) p(z_{ik}|w_{ik}, m_k) \prod_{j=1}^{2S} p(c_{ijk}|z_{ik})$$

[2] Lowercase *lmc* refers to the latent variable in the uppercase *LMC* graphical model.

[3] We use *pseudo* because the LMC is a latent variable model, not a conventional generative model. As with the Skip-Gram model, due to the re-use of data (center and context words), we cannot use LMC to generate new text, but we can specify an objective function on existing data.

3.4. Distributions

We assume the following model distributions: $m_k \sim \text{Cat}(\gamma)$, $w_{ik} \sim \text{Cat}(\alpha)$, and $z_{ik}|w_{ik}, m_k \sim N(nn(w_{ik}, m_k; \theta))$. $nn(w_{ik}, m_k; \theta)$ denotes a neural network that outputs isotropic Gaussian parameters. $p(c_{ijk}|z_{ik})$ is simply a normalized function of fixed parameters (θ) and z_{ik}. We choose a form that resembles Bayes' Rule and compute the ratio of the joint to the marginal:

$$p(c_{ijk}|z_{ik}) = \frac{\sum_m p(z_{ik}|c_{ijk}, m)p(m|c_{ijk})p(c_{ijk})}{\sum_m \sum_c p(z_{ik}|c, m)p(m|c)p(c)} \tag{1}$$

We marginalize over metadata and factorize to include $p(z_{ik}|c_{ijk}, m)$, which shares parameters θ with $p(z_{ik}|w_{ik}, m_k)$. The prior over meaning is modeled as in Sohn et al. (2015). $p(m|c)$ and $p(c)$ are defined by corpus statistics. Therefore, the set of parameters that define $p(z_{ik}|w_{ik}, m_k)$ completely determines $p(c_{ijk}|z_{ik})$, making for efficient inference.

4. Inference

Ideally, we would like to make posterior inference on lmcs given observed variables. For one center word w_{ik}, this requires modeling

$$p(z_{ik}|m_k, w_{ik}, \boldsymbol{c_{ik}}) = \frac{p(z_{ik}, m_k, w_{ik}, \boldsymbol{c_{ik}})}{\int p(z_{ik}, m_k, w_{ik}, \boldsymbol{c_{ik}}) d_{z_{ik}}}$$

Unfortunately, the posterior is intractable because of the integral. Instead, we use variational Bayes to minimize the KL-Divergence (KLD) between an amortized variational family and the posterior:

$$\min_{\phi,\theta} D_{KL}\Big(Q_\phi(Z|M, W, C)||P_\theta(Z|M, W, C)\Big)$$

4.1. Deriving the Final Objective

At a high level, we factorize distributions (A.3.1) and then derive an analytical form of the KLD to arrive at a final objective (4.1.1). We then explain the use of approximate bounds for efficiency: the likelihood with negative sampling (4.1.2), and the KLD between the variational distribution and an unbiased mixture estimation (4.1.3).

4.1.1. Final Objective

To avoid high variance, we derive the analytical form of the objective function, rather than optimize with score gradients (Ranganath et al., 2014; Schulman et al., 2015). For each center word, the loss function we minimize is:

$$\begin{aligned} L_{\phi,\theta}(m_k, w_{ik}, \boldsymbol{c_{ik}}) = \sum_{j=1}^{2S} \max \Bigg(0, \\ D_{KL}\Big(q_{ik}|| \sum_m p_\theta(z_{ik}|c_{ijk}, m)\beta_{m|c_{ijk}}\Big) \\ - D_{KL}\Big(q_{ik}|| \sum_m p_\theta(z_{ik}|\tilde{c}, m)\beta_{m|\tilde{c}}\Big)\Bigg) \\ + D_{KL}\Big(q_{ik}||p_\theta(z_{ik}|m_k, w_{ik})\Big) \end{aligned} \tag{2}$$

where q_{ik} denotes $q_\phi(z_{ik}|m_k, w_{ik}, \boldsymbol{c_{ik}})$. $\tilde{c}$ represents a negatively sampled word. We denote the empirical likelihoods of metadata given a context / negatively sampled word as $\beta_{m|c_{ijk}}$ / $\beta_{m|\tilde{c}}$. Intuitively, the objective rewards reconstruction of context words through the approximate posterior while encouraging it not to stray too far from the center word's marginal meaning across metadata. We include the full derivation in A.3.

4.1.2. Negative Sampling

As in the BSG model, we use negative sampling as an efficient lower bound of the marginal likelihood from Equation 1. $\tilde{c}$ is sampled from the empirical vocabulary distribution $p(\tilde{c})$ to construct an unbiased estimate for $E_{\tilde{c}}\Big[\sum_m p_\theta(z_{ik}|\tilde{c}, m)\beta_{m|\tilde{c}}\Big]$. Finally, we transform the likelihood into a hard margin to bound the loss and stabilize training.

4.1.3. KL-Divergence for Mixtures

The objective requires computing the KLD between a Gaussian (q_{ik}) and a Gaussian mixture ($\sum_m p_\theta(z|c,m)\beta_{m|c}$). To avoid computing the full marginal, for both context words and negatively sampled words, we sample ten metadata using the appropriate empirical distribution: $\beta_{m|c_{ijk}}$ and $\beta_{m|\tilde{c}}$, respectively. Using this unbiased sample of mixtures, we form an upper bound for the KLD between the variational family and an unbiased mixture estimation (Hershey and Olsen, 2007): $D_{KL}(f||g) \leq \sum_{a,b}\pi_a\omega_b D_{KL}(f_a||g_b)$. π_a is the mixture weight of f and ω_b is the mixture weight of g. f is the variational distribution formed by a single Gaussian and g is the mixture of interest. Thus, the upper-bound is simply the weighted sum of the KLD between the variational distribution and each mixture component.

4.2. Training Algorithm

Algorithm 1: LMC Training Procedure

```
Randomly initialize parameters: $\phi, \theta$
while not converged do
    Sample mini-batch $m_k, w_{ik}, \boldsymbol{c_{ik}} \sim D$
    $\delta \leftarrow \nabla_{\phi,\theta} L_{\phi,\theta}(m_k, w_{ik}, \boldsymbol{c_{ik}})$
    $\phi, \theta \leftarrow$ Update using gradient $\delta$
end
```

The training procedure samples a center word, context word sequence, and metadata from the data distribution and minimizes the loss function from Equation 2 with stochastic gradient descent. In Algorithm 1, we jointly update the variational family and model parameters, ϕ and θ respectively.

5. Neural Networks

The LMC model requires modeling two Gaussian distributions, $q_\phi(z_{ik}|m_k, w_{ik}, \boldsymbol{c_{ik}})$ and $p_\theta(z_{ik}|c_{ijk}, m)$. We parametrize both with neural networks, but any black-box function suffices. We refer to q_ϕ as the `variational network` and p_θ as the `model network`.

5.1. Variational Network (q_ϕ)

The variational network accepts a center word w_{ik}, metadata m_k, and a sequence of context words $\boldsymbol{c_{ik}}$, and outputs isotropic Gaussian parameters: a mean vector μ_q and variance scalar σ_q. Then, $q_\phi \sim N(\mu_q, \sigma_q)$. At a high level, we encode words with a bi-LSTM (Graves et al., 2005), summarize the sequence with metadata-specific attention, and then learn a gating function to selectively combine evidence. A.4 contains the full specification.

5.2. Model Network (p_θ)

The model network accepts a word w_{ik} and metadata m_k and projects them onto a higher dimension with embedding matrix R. $R_{w_{ik}}$ and R_{m_k} are combined: $h = ReLU(W_{model}([R_{w_{ik}}; R_{m_k}]) + b)$. The hidden state h is then separately projected to produce a mean vector μ_p and variance scalar σ_p. Then, $p_\theta \sim N(\mu_p, \sigma_p)$.

6. Experimental Setup

We pre-train the LMC model and all baselines on unlabeled MIMIC-III notes and compare zero-shot performance on three acronym expansion datasets. Because we consider the zero-shot scenario, we restrict ourselves to pre-trained contextualized embedding models without fine-tuning. Out of fidelity to the data, we do not adjust the natural class imbalances. We explicitly test each model's ability to handle rare expansions, for which shared statistical strength from metadata may be critical. All models receive the same local word context, yet only two models (MBSGE, LMC) receive section header metadata. We include full details for Section 6 in A.5.

6.1. Pre-Training

MIMIC-III contains de-identified clinical records from patients admitted to Beth Israel Deaconess Medical Center (Johnson et al., 2016). It comprises two million documents spanning sixteen note types, from discharge summaries to radiology reports. Section headers are extracted through regular expressions. We pre-train all models for five epochs in PyTorch (Paszke et al., 2017) and report results on one test set using the others for validation.

6.2. Evaluation Data

It is difficult to acquire annotated data for clinical acronym expansion, especially with relevant metadata. One of the few publicly available datasets with section header annotations is the Clinical Abbreviation Sense Inventory (**CASI**) dataset (Moon et al., 2014). Human annotators assign expansion labels to a set of 74 clinical abbreviations in context. The authors remove ambiguous examples (based on local word context alone) before publishing the data. Our experimental test set comprises 27,209 examples across 41 unique acronyms and 150 expansions.

To evaluate across a range of institutions, as well as consider all examples (even ambiguous), we use the acronym sense inventory from CASI to construct two new synthetic datasets via reverse substitution (RS). RS involves replacing long form expansions with their corresponding short form and then assigning the original expansion as the target label (Finley et al., 2016). 44,473 tuples of (short form context, section header, target long form) extracted from MIMIC comprise the **MIMIC RS** dataset. The second RS dataset consists of 22,163 labeled examples from a corpus of 150k ICU/CCU notes collected between 2005 and 2015 at the Columbia University Irving Medical Center (**CUIMC**). For each RS datasest, we draw at most 500 examples per acronym-expansion pair. For non-MIMIC datasets, when a section does not map to one in MIMIC, we choose the closest corollary.

6.3. Baselines

Dominant & Random Class. Acronym expansion datasets are highly imbalanced. Dominant class accuracy, then, tends to be high and is useful for putting metrics into perspective. Random performance provides a crude lower bound.

Section Header MLE. To isolate the discriminative power of section headers, we include a simple baseline which selects LFs based on $p(LF|section) \propto p(section|LF)$. We compute $p(section|LF) = \frac{C(section,LF)}{C(LF)}$ on held-out data.

Bayesian Skip-Gram (BSG). We implement our own version of the BSG model so that it uses the same `variational network` architecture as the LMC, with the exception that metadata is unavailable.

Metadata BSG Ensemble (MBSGE). To isolate the added-value of metadata, we devise an ensembled BSG. MBSGE maintains an identical optimization procedure with the exception that it treats metadata and center words as interchangeable observed variables. During training, center words are randomly replaced with metadata, which take on the function of a center word. For evaluation, we average ensemble the contextualized embeddings from metadata and center word. We train on two metadata types: section headers and note type, but for experiments, based on available data, we only use headers.

ELMo. We use the AllenNLP implementation with default hyperparameters for the Transformer-based version (Gardner et al., 2018; Peters et al., 2018b). We pre-train the model for five epochs with a batch size of 3,072. We found optimal performance by taking the sequence-wise mean rather than selecting the hidden state from the SF index.

BERT. Due to compute limitations, we rely on the publicly available Clinical

BioBERT for evaluation (Alsentzer et al., 2019; Lee et al., 2020). We access the pre-trained model through the Hugging Face Transformer library (Wolf et al., 2019). The weights were initialized from BioBERT (introduces Pubmed articles) before being fine-tuned on the MIMIC-III corpus. We experimented with many pooling configurations and found that taking the average of the mean and max from the final layer performed best on a validation set. Another ClinicalBERT uses this configuration (Huang et al., 2019).

6.4. Task Definition

We rank each candidate acronym expansion (LF) by measuring similarity between its context-independent representation and the contextualized acronym representation. Table 1 shows the ranking functions we used. $ELMO_{avg}$ represents the mean of final hidden states. For the LMC scoring function, $\sum_m p(z|LF_k, m)\beta_{m|LF_k})$ represents the smoothed marginal distribution of a word (or phrase) over metadata (as detailed in A.9).

Table 1: LF_k represents the k^{th} LF.

Model	**Ranking Function**
BERT	$Cosine(BERT_{avg}^{max}(SF; \boldsymbol{c}), BERT_{avg}^{max}(LF_k))$
ELMo	$Cosine(ELMO_{avg}(SF; \boldsymbol{c}), ELMO_{avg}(LF_k))$
BSG	$D_{KL}(q(z \mid SF, \boldsymbol{c}) \| p(z \mid LF_k))$
MBSGE	$D_{KL}(Avg_{x \in \{SF, m\}}(q(z \mid x, \boldsymbol{c})) \| p(z \mid LF_k))$
LMC	$D_{KL}(q(z \mid SF, m, \boldsymbol{c}) \| \sum_m p(z \mid LF_k, m)\beta_{m \mid LF_k})$

7. Results

7.1. Classification Performance

Recent work has shown that randomness in pre-training contextualized LMs can lead to large variance on downstream tasks (Dodge et al., 2020). For robustness, then, we pre-train five separate weights for each model class and report aggregate results. Tables 2 and 3 show mean statistics for each model across five pre-training runs. In A.6.1, we show best/worst performance, as well as bootstrap each test set to generate confidence intervals (A.6.2). These additional experiments add robustness and reveal de minimus variance between LMC pre-training runs and between bootstrapped test sets for a single model. Our main takeaways are:

Metadata. The MBSGE and LMC models materially outperform non-metadata baselines, which suggests that metadata is complementary to local word context for the task.

LMC Robust Performance. The LMC outperforms all baselines and exhibits very low variance across pre-training runs. Given the same input and very similar parameters as MBSGE, the LMC model appears useful beyond the addition of a helpful feature.

Dataset Comparison. Unsurprisingly, performance is best on the MIMIC RS dataset because all models are pre-trained on MIMIC notes. While CUIMC and CASI are in-domain, there is minor performance degradation from the transfer.

Lower CASI Spread. The LMC performance gains are less pronounced on the public CASI dataset. CASI was curated to only include examples whose expansions could be unambiguously deduced from local context by humans. Hence, the relative explanatory power of metadata is likely dampened.

Poor BERT, ELMo Performance. BERT / ELMo underperform across datasets. They are optimized to assign high probability to masked or next-word tokens, not to align embedded representations. For our zero-shot use case, then, they may represent suboptimal pre-training objectives. Meanwhile, the BSG, MBSGE, and LMC models are trained to align context-dependent representations (`variational network`) with corresponding context-independent representations (`model network`). For evaluation, we simply replace context words with candidate LFs.

Table 2: Mean across 5 pre-training runs. NLL is neg log likelihood, W/M weighted/macro.

	MIMIC				CUIMC				CASI			
Model	NLL	Acc	W F1	M F1	NLL	Acc	W F1	M F1	NLL	Acc	W F1	M F1
BERT	1.36	0.40	0.40	0.33	1.41	0.37	0.33	0.28	1.23	0.42	0.38	0.23
ELMo	1.33	0.58	0.61	0.53	1.38	0.58	0.60	0.49	1.21	0.55	0.56	0.38
BSG	1.28	0.57	0.59	0.52	9.04	0.58	0.58	0.46	0.99	0.64	0.64	0.41
MBSGE	1.07	0.65	0.67	0.59	6.16	0.64	0.64	0.52	0.88	0.70	0.70	0.46
LMC	**0.81**	**0.74**	**0.78**	**0.69**	**0.90**	**0.69**	**0.68**	**0.57**	**0.79**	**0.71**	**0.73**	**0.51**

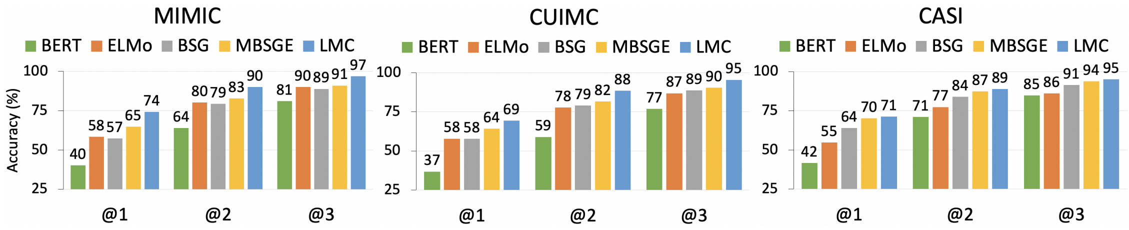


Figure 3: Average accuracy @K across 5 pre-training runs.

Non-Parametric. Random/dominant accuracy is 27/42%, 26/47%, and 31/78% for MIMIC, CUIMC, and CASI. Section information alone proves very discriminative on MIMIC (85% accuracy for Section Header MLE), but, given the sparse distribution, it severely overfits. On CASI/CUIMC, the accuracy plummets to 48/46% and macro F1 to 35/33%. While relevant, generalization requires distributional header representations.

7.2. Qualitative Analysis

7.2.1. Word-Metadata Gating

Inside the `variational network`, the network learns a weighted average of metadata and word level representations. We examine instances where more weight is placed on local acronym context vis-a-vis section header, and vice versa. Table 3 shows that shorter sections with limited topic diversity (e.g., "Other ICU Medications") are assigned greater relative weight. The network selectively relies on each source based on relative informativeness.

The gating function enables manual interpolation between local context and metadata to measure smoothness in word meaning transitions. We select three sections which a priori we associate with expansions of the acronym MG: "Discharge Medications" with milligrams, "Imaging" with myasthenia gravis, and "Review of Systems" with magnesium (deficiency). We compute the lmc conditioned on "MG" and each section m, ranking LFs by taking the softmax over $-D_{KL}(q(z|MG, m, c_{\emptyset}||p(z|LF, m_{\emptyset})))$, where $c_{\emptyset}$ and $m_{\emptyset}$ denote null values. Figure 4 shows a gradual transition between meanings, suggesting the `variational network` is a smooth function approximator.

7.2.2. lmcs as Word Senses

A guiding principle behind the LMC model centers on the power of metadata to disambiguate polysemous words. We choose the word "history" and enumerate five diverse types of patient history: smoking, depression, diabetes, cholesterol, and heart. Then, we examine the proximity of lmcs for the target

Table 3: `variational network` gating function weights.

Target Label	**Context Window**	**Section Weight**
patent ductus arteriosis	*Hospital Course*: echocardiogram showed normal heart structure with **PDA** hemodynamically significant	0.12
pulmonary artery	*Tricuspid Valve*: physiologic tr pulmonic valve **PA** physiologic normal pr	0.21
no acute distress	*General Appearance*: well nourished **NAD**	0.38
morphine sulfate	*Other ICU Medications*: **MS** $\langle digit \rangle$ pm	0.46

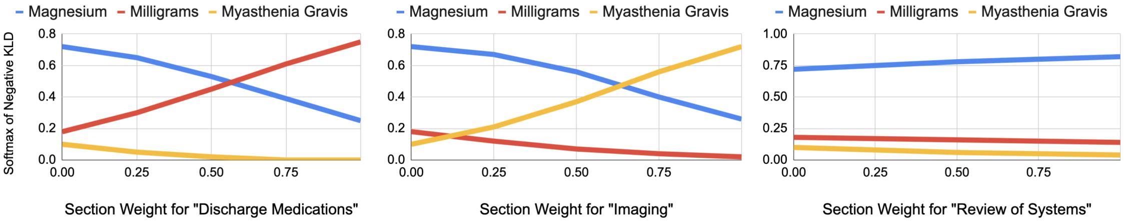


Figure 4: The latent sense distribution changes when manually interpolating the `variational network` weight between the word "MG" & different section headers.

Table 4: Conditional latent meaning: history

Section	**Most Similar Words**
Past Medical History	depression, diabetes
Social History	smoking, depression
Family History	depression, smoking
Glycemic Control	cholesterol, diabetes
Left Ventricle	heart, depression
Nutrition	diabetes, cholesterol

Table 5: Section header embeddings.

Section	**Nearest Neighbors**
Allergies	Social History, Prophylaxis, Disp
Chief Complaint	Reason, Family History, Indication
History of Present Illness	HPI, Past Medical History, Total Time Spent
Meds on Admission	Discharge Medications, Other Medications, Disp
Past Medical History	HPI, Social History, History of Present Illness

word under relevant section headers and compare to the expected representations of the five types of patient history. Section headers have a largely positive impact on word meanings (Table 4), especially for generic words with large prior variances like "history".

7.2.3. Clustering Section Headers

In Table 5, we select five prominent headers and measure cosine proximity of embeddings learned by the `variational network`[4]. In most cases, the results are meaningful, even uncovering a section acronym: "HPI" for "History of Present Illness".

8. Conclusion

We target a key problem in clinical text, introduce a helpful feature, and present a Bayesian

[4] No difference from using `model network`.

solution that works well on the task. More generally, the LMC model presents a principled, efficient approach for incorporating metadata into language modeling.

9. Citations and Bibliography

Acknowledgments

We thank Arthur Bražinskas, Rajesh Ranganath, and the reviewers for their constructive, thoughtful feedback. This work was supported by NIGMS award R01 GM114355 and NCATS award U01 TR002062.

References

Emily Alsentzer, John R Murphy, Willie Boag, Wei-Hung Weng, Di Jin, Tristan Naumann, and Matthew McDermott. Publicly available clinical bert embeddings. *arXiv preprint arXiv:1904.03323*, 2019.

Ben Athiwaratkun and Andrew Wilson. Multimodal word distributions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1645–1656, 2017.

Ben Athiwaratkun, Andrew Wilson, and Anima Anandkumar. Probabilistic FastText for multi-sense word embeddings. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1–11, 2018.

Sergey Bartunov, Dmitry Kondrashkin, Anton Osokin, and Dmitry Vetrov. Breaking sticks and ambiguities with adaptive skip-gram. In *Artificial Intelligence and Statistics*, pages 130–138, 2016.

David M Blei, Andrew Y Ng, and Michael I Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

Olivier Bodenreider. The unified medical language system (UMLS): integrating biomedical terminology. *Nucleic Acids Research*, 32(suppl_1):D267–D270, 2004.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017.

Samuel R. Bowman, Ellie Pavlick, Edouard Grave, Benjamin Van Durme, Alex Wang, Jan Hula, Patrick Xia, Raghavendra Pappagari, R. Thomas McCoy, Roma Patel, Najoung Kim, Ian Tenney, Yinghui Huang, Katherin Yu, Shuning Jin, and Berlin Chen. Looking for ELMo's friends: Sentence-level pretraining beyond language modeling, 2019. URL https://openreview.net/forum?id=Bkl87h09FX.

Arthur Bražinskas, Serhii Havrylov, and Ivan Titov. Embedding words as distributions with a Bayesian skip-gram model. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1775–1789, 2018.

Jose Camacho-Collados and Mohammad Taher Pilehvar. From word to sense embeddings: A survey on vector representations of meaning. *Journal of Artificial Intelligence Research*, 63(1): 743–788, 2018.

Rajarshi Das, Manzil Zaheer, and Chris Dyer. Gaussian LDA for topic models with word embeddings. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 795–804, 2015.

D Demner-Fushman and Noémie Elhadad. Aspiring to unintended consequences of natural language processing: a review of recent developments in clinical and consumer-generated text processing. *Yearbook of Medical Informatics*, 25(01):224–233, 2016.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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, Volume 1 (Long and Short Papers)*, pages 4171–4186, 2019.

Adji B Dieng, Francisco JR Ruiz, and David M Blei. Topic modeling in embedding spaces. *arXiv preprint arXiv:1907.04907*, 2019.

Jesse Dodge, Gabriel Ilharco, Roy Schwartz, Ali Farhadi, Hannaneh Hajishirzi, and Noah Smith. Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping. *arXiv preprint arXiv:2002.06305*, 2020.

Gregory P Finley, Serguei VS Pakhomov, Reed McEwan, and Genevieve B Melton. Towards comprehensive clinical abbreviation disambiguation using machine-labeled training data. In *AMIA Annual Symposium Proceedings*, page 560, 2016.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer. AllenNLP: A deep semantic natural language processing platform. In *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*, pages 1–6, 2018.

Alex Graves, Santiago Fernández, and Jürgen Schmidhuber. Bidirectional LSTM networks for improved phoneme classification and recognition. In *International Conference on Artificial Neural Networks*, pages 799–804. Springer, 2005.

John R Hershey and Peder A Olsen. Approximating the kullback leibler divergence between gaussian mixture models. In *2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP'07*, pages IV–317, 2007.

Kexin Huang, Jaan Altosaar, and Rajesh Ranganath. Clinicalbert: Modeling clinical notes and predicting hospital readmission. *arXiv preprint arXiv:1904.05342*, 2019.

Qiao Jin, Bhuwan Dhingra, William W Cohen, and Xinghua Lu. Probing biomedical embeddings from language models. *arXiv preprint arXiv:1904.02181*, 2019a.

Qiao Jin, Jinling Liu, and Xinghua Lu. Deep contextualized biomedical abbreviation expansion. *arXiv preprint arXiv:1906.03360*, 2019b.

Alistair EW Johnson, Tom J Pollard, Lu Shen, H Lehman Li-wei, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. MIMIC-III, a freely accessible critical care database. *Scientific Data*, 3:160035, 2016.

Mahesh Joshi, Serguei Pakhomov, Ted Pedersen, and Christopher G Chute. A comparative study of supervised learning as applied to acronym expansion in clinical reports. In *AMIA annual symposium proceedings*, volume 2006, page 399, 2006.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Hugo Larochelle and Stanislas Lauly. A neural autoregressive topic model. In *Advances in Neural Information Processing Systems*, pages 2708–2716, 2012.

Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In *International Conference on Machine Learning*, pages 1188–1196, 2014.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240, 2020.

Yoav Levine, Barak Lenz, Or Dagan, Dan Padnos, Or Sharir, Shai Shalev-Shwartz, Amnon Shashua, and Yoav Shoham. Sensebert: Driving some sense into bert. *arXiv preprint arXiv:1908.05646*, 2019.

Irene Li, Michihiro Yasunaga, Muhammed Yavuz Nuzumlalı, Cesar Caraballo, Shiwani Mahajan, Harlan Krumholz, and Dragomir Radev. A neural topic-attention model for medical term abbreviation disambiguation. *arXiv preprint arXiv:1910.14076*, 2019.

Shaohua Li, Tat-Seng Chua, Jun Zhu, and Chunyan Miao. Generative topic embedding: a continuous representation of documents. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 666–675, 2016.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

Edward Loper and Steven Bird. NLTK: The natural language toolkit. In *Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics*, pages 63–70, 2002.

SM Meystre, GK Savova, KC Kipper-Schuler, and JF Hurdle. Extracting information from textual documents in the electronic health record: a review of recent research. *Yearbook of Medical Informatics*, pages 128–44, 2008.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013a.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119, 2013b.

George A Miller. *WordNet: An electronic lexical database.* MIT press, 1998.

Yasumasa Miyamoto and Kyunghyun Cho. Gated word-character recurrent language model. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1992–1997, 2016.

Sungrim Moon, Serguei Pakhomov, Nathan Liu, James O Ryan, and Genevieve B Melton. A sense inventory for clinical abbreviations and acronyms created using clinical notes and medical dictionary resources. *Journal of the American Medical Informatics Association*, 21(2):299–307, 2014.

Arvind Neelakantan, Jeevan Shankar, Alexandre Passos, and Andrew McCallum. Efficient non-parametric estimation of multiple embeddings per word in vector space. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1059–1069, 2014.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in PyTorch. In *31st Conference on Neural Information Processing Systems (NIPS 2017)*, 2017.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. 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 2227–2237, 2018a.

Matthew Peters, Mark Neumann, Luke Zettlemoyer, and Wen-tau Yih. Dissecting contextual word embeddings: Architecture and representation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1499–1509, 2018b.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI Blog*, 1 (8):9, 2019.

Rajesh Ranganath, Sean Gerrish, and David Blei. Black box variational inference. In *Artificial Intelligence and Statistics*, pages 814–822, 2014.

John Schulman, Nicolas Heess, Theophane Weber, and Pieter Abbeel. Gradient estimation using stochastic computation graphs. In *Advances in Neural Information Processing Systems*, pages 3528–3536, 2015.

Marta Skreta, Aryan Arbabi, Jixuan Wang, and Michael Brudno. Training without training data: Improving the generalizability of automated medical abbreviation disambiguation. *arXiv preprint arXiv:1912.06174*, 2019.

Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. In *Advances in neural information processing systems*, pages 3483–3491, 2015.

Nitish Srivastava, Ruslan R Salakhutdinov, and Geoffrey E Hinton. Modeling documents with deep boltzmann machines. In *Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence (UAI2013)*, 2013.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of Machine Learning Research*, 15(1):1929–1958, 2014.

Emma Strubell, Ananya Ganesh, and Andrew McCallum. Energy and policy considerations for deep learning in nlp. *arXiv preprint arXiv:1906.02243*, 2019.

Fei Tian, Hanjun Dai, Jiang Bian, Bin Gao, Rui Zhang, Enhong Chen, and Tie-Yan Liu. A probabilistic model for learning multi-prototype word embeddings. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 151–160, 2014.

Hilary Townsend. Natural language processing and clinical outcomes: the promise and progress of nlp for improved care. *Journal of AHIMA*, 84(2):44–45, 2013.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

Luke Vilnis and Andrew McCallum. Word representations via gaussian embedding. *arXiv preprint arXiv:1412.6623*, 2014.

Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103, 2008.

L Weed. Medical records that guide and teach. *New England Journal of Medicine*, 278:593–600, 1968.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. Transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*, 2019.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. XLNet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pages 5754–5764, 2019.

Dani Yogatama, Cyprien de Masson d'Autume, Jerome Connor, Tomas Kocisky, Mike Chrzanowski, Lingpeng Kong, Angeliki Lazaridou, Wang Ling, Lei Yu, Chris Dyer, et al. Learning and evaluating general linguistic intelligence. *arXiv preprint arXiv:1901.11373*, 2019.

Ming Zhu, Busra Celikkaya, Parminder Bhatia, and Chandan K Reddy. LATTE: Latent type modeling for biomedical entity linking. In *AAAI Conference*, 2020.

Appendix A. Appendix

A.1. Future Work

We hope the LMC framework and code base encourages research into metadata-based language modeling: **(1) New domains.** The LMC can be applied to any domain which discrete metadata provides informative contextual clues (e.g., document categories, sections, document ids). **(2) Linguistic Properties.** A unique feature of the LMC is the ability to represent words as marginal distributions over metadata, and vice versa (as detailed in A.8). We encourage exploration into its linguistic implications. **(3) Metadata Skip-Gram.** Depending on the choice of metadata, the LMC model could be expanded to draw context metadata from a center metadata. This might capture metadata-level entailment. **(4) Calibration.** Modeling words and metadata as Gaussian densities can facilitate analysis to connect variance to model uncertainty, instrumental in real-world applications with user feedback. **(5) Sub-Words.** In morphologically rich languages, subword information has been shown to be highly effective for sharing statistical strength across morphemes (Bojanowski et al., 2017). Probabilistic FastText may provide a blueprint for incorporating subwords into LMC (Athiwaratkun et al., 2018).

A.2. Metadata Pseudo Document

For our experiments, metadata is comprised of the concatenation of the body of every section header across the corpus. Yet, when computing context windows, we do not combine text from different physical documents. Please see Figure 5 for a toy example.

A.3. Full Derivations

A.3.1. Factorize & Reduce

After factorizing the model posterior and variational distribution, we can push the integral inside the summation and integrate out latent variables that are independent:

$$\sum_{i,k} \int q_{\phi}(z_{ik}|m_k, w_{ik}, \boldsymbol{c_{ik}}) \log \frac{q_{\phi}(z_{ik}|m_k, w_{ik}, \boldsymbol{c_{ik}})}{p_{\theta}(z_{ik}|m_k, w_{ik}, \boldsymbol{c_{ik}})} d_{z_{ik}} \tag{3}$$

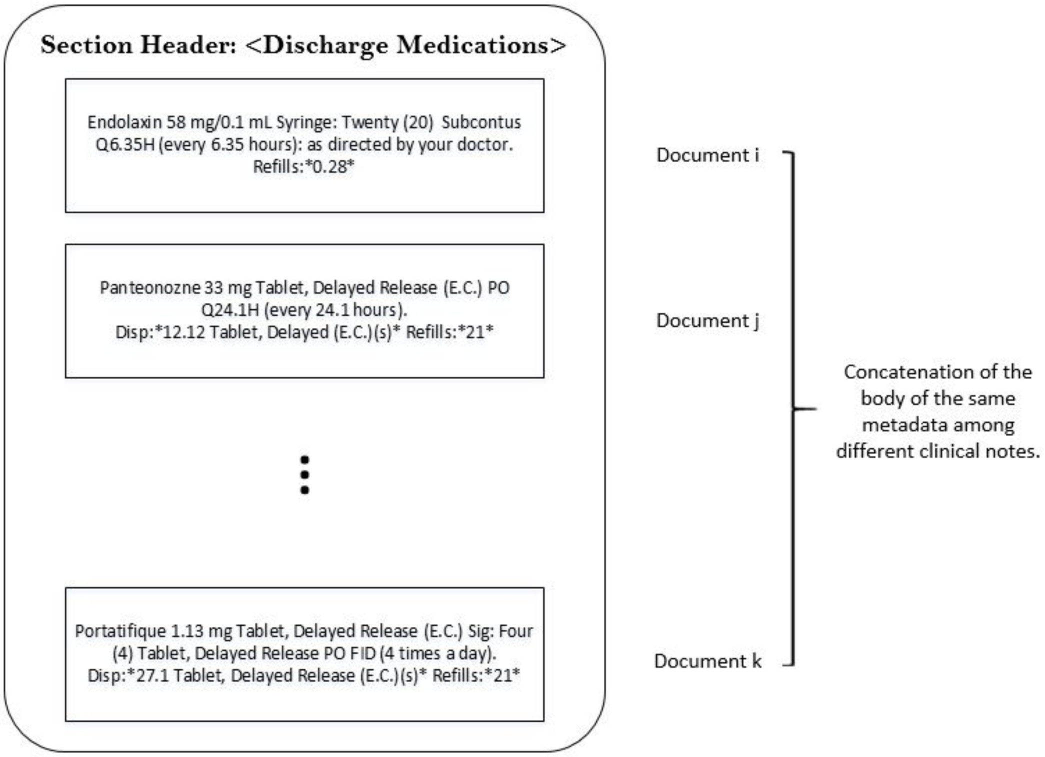
The integral defines a KL measure between individual latent variables, which can be expressed as

$$|W| \frac{1}{|W|} \sum_{i,k} E_{q_{ik}} \Big[\log \frac{q_{\phi}(z_{ik}|m_k, w_{ik}, \boldsymbol{c_{ik}})}{p_{\theta}(z_{ik}|m_k, w_{ik}, \boldsymbol{c_{ik}})} \Big] \tag{4}$$

where $|W|$ represents the corpus word count. Dividing and multiplying by $|W|$ does not change the result:

$$E_{\hat{p}} \Bigg[D_{KL} \Big(q_{ik} || p_{ik} \Big) \Bigg] \tag{5}$$

We ignore $|W|$, as it does not affect the optimization, and denote the amortized variational distribution, model posterior, and the empirical uniform distribution over center words in the corpus as q_{ik}, p_{ik}, and $\hat{p}$, respectively.



Section Header: <Discharge Medications>

Endolaxin 58 mg/0.1 mL Syringe: Twenty (20) Subcontus Q6.35H (every 6.35 hours): as directed by your doctor. Refills:*0.28*

Panteonozne 33 mg Tablet, Delayed Release (E.C.) PO Q24.1H (every 24.1 hours). Disp:*12.12 Tablet, Delayed (E.C.)(s)* Refills:*21*

Portatifique 1.13 mg Tablet, Delayed Release (E.C.) Sig: Four (4) Tablet, Delayed Release PO FID (4 times a day). Disp:*27.1 Tablet, Delayed Release (E.C.)(s)* Refills:*21*

Document i
Document j
Document k
Concatenation of the body of the same metadata among different clinical notes.

Figure 5: Metadata Pseudo Document for *DISCHARGE MEDICATIONS*.

A.3.2. LMC Objective

In the main manuscript, we outline the steps involved to arrive at the variational objective. Here, we break it down into a more complete derivation. Because the posterior of the LMC model is intractable, we use variational Bayes and minimize the KLD between the variational distribution and the model posterior:

$$\min D_{KL}\Big(Q(Z|M,W,C)||P(Z|M,W,C)\Big) \tag{6}$$

KL-Divergence can also be expressed in expected value form:

$$\min E_Q\Big[\log\frac{Q(Z|M,W,C)}{P(Z|M,W,C)}\Big] \tag{7}$$

The expectation can be re-written in the integral form as follows:

$$\min\int log\frac{Q(Z|M,W,C)}{P(Z|M,W,C)}Q(Z|M,W,C)d_Z \tag{8}$$

Using the independence assumption of the latent random variables, we can factor Q and P as follows:

$$\min \int ... \int \log \frac{\prod_{i,k} q(z_{ik}|m_k, w_{ik}, \boldsymbol{c_{ik}})}{\prod_{i,k} p(z_{ik}|m_k, w_{ik}, \boldsymbol{c_{ik}})} \prod_{i,k} q(z_{ik}|m_k, w_{ik}, \boldsymbol{c_{ik}}) d_{z_{ik}} \tag{9}$$

Taking the product out of the logarithm yields

$$\min \int ... \int \sum_{i,k} log \frac{q(z_{ik}|m_k, w_{ik}, \boldsymbol{c_{ik}})}{p(z_{ik}|m_k, w_{ik}, \boldsymbol{c_{ik}})} \prod_{i,k} q(z_{ik}|m_k, w_{ik}, \boldsymbol{c_{ik}}) d_{z_{ik}} \tag{10}$$

We can push the integral inside the summation by integrating independent latent variables out:

$$\min \sum_{i,k} \int \log \frac{q(z_{ik}|m_k, w_{ik}, \boldsymbol{c_{ik}})}{p(z_{ik}|m_k, w_{ik}, \boldsymbol{c_{ik}})} q(z_{ik}|m_k, w_{ik}, \boldsymbol{c_{ik}}) d_{z_{ik}} \tag{11}$$

Dividing the summation by the number of words in the corpus defines an expectation over the KL-Divergence for each independent latent variable. Here, $|W|$ denotes the number of words in the corpus. Multiplying the above expression by $|W|$ and dividing by $|W|$ doesn't change the result. Thus,

$$\min |W| \frac{1}{|S|} \sum_{i,k} E_{q_{ik}} \Big[\log \frac{q(z_{ik}|m_k, w_{ik}, \boldsymbol{c_{ik}})}{p(z_{ik}|m_k, w_{ik}, \boldsymbol{c_{ik}})} \Big] \tag{12}$$

$\frac{1}{|W|} \sum_{i,k}$ defines an expectation over the observed data. Therefore, we can write the above expression as

$$\min E_{m_k, w_{ik}, \boldsymbol{c_{ik}} \sim D} \Bigg[E_{q_{ik}} \Big[\log \frac{q(z_{ik}|m_k, w_{ik}, \boldsymbol{c_{ik}})}{p(z_{ik}|m_k, w_{ik}, \boldsymbol{c_{ik}})} \Big] \Bigg] \tag{13}$$

Here the expression $m_k, w_{ik}, \boldsymbol{c_{ik}} \sim D$ denotes sampling observed variables of document, center word and context words from the data distribution. We ignore $|W|$ as it does not affect the optimization:

$$\min E_{m_k, w_{ik}, \boldsymbol{c_{ik}} \sim D} \Bigg[D_{KL} \Big(q(z_{ik}|m_k, w_{ik}, \boldsymbol{c_{ik}}) || p(z_{ik}|m_k, w_{ik}, \boldsymbol{c_{ik}}) \Big) \Bigg] \tag{14}$$

The above expression represents the final objective function. To optimize, we sample $m_k, w_{ik}, \boldsymbol{c_{ik}} \sim D$ and minimize the KL-Divergence between q and p. Here D represents the distribution of data from the corpus, which we assume is uniform across observed metadata and words.

A.3.3. Analytical Form of KL-Divergence

One can approximate KL-Divergence by sampling. Yet, such an estimate has high variance. To avoid this, we derive the analytical form of the objective function. From Section A.3.2, we seek to minimize the following objective function:

$$D_{KL} \Big(q_\phi(z_{ik}|m_k, w_{ik}, \boldsymbol{c_{ik}}) || p_\theta(z_{ik}|m_k, w_{ik}, \boldsymbol{c_{ik}}) \Big) \tag{15}$$

The above equation can be expressed as

$$E_{q_{ik}}\Big[\log q_\phi(z_{ik}|m_k, w_{ik}, \boldsymbol{c_{ik}}) - \log p_\theta(z_{ik}, m_k, w_{ik}, \boldsymbol{c_{ik}})\Big] + \log p(m_k, w_{ik}, \boldsymbol{c_{ik}}) \tag{16}$$

We can factorize $p_\theta(z_{ik}, m_k, w_{ik}, \boldsymbol{c_{ik}})$ using the model family definition

$$E_{q_{ik}}\Big[\log q_\phi(z_{ik}|m_k, w_{ik}, \boldsymbol{c_{ik}}) - \log p(m_k)p(w_{ik})p_\theta(z_{ik}|w_{ik}, m_k)\prod_{j=1}^{2S} p_\theta(c_{ijk}|z_{ik})\Big] + \log p(m_k, w_{ik}, \boldsymbol{c_{ik}}) \tag{17}$$

Since, $p(m_k, w_{ik}, \boldsymbol{c_{ik}}) = p(\boldsymbol{c_{ik}}|m_k, w_{ik})p(w_{ik})p(m_k)$, we can re-write Equation 17 as

$$\begin{aligned} E_{q_{ik}}\Big[\log q_\phi(z_{ik}|m_k, w_{ik}, \boldsymbol{c_{ik}}) - \log p(m_k) - \log p(w_{ik}) - \log p_\theta(z_{ik}|w_{ik}, m_k) \\ - \sum_{j=1}^{2S} \log p_\theta(c_{ijk}|z_{ik})\Big] + \log p(\boldsymbol{c_{ik}}|m_k, w_{ik}) + \log p(m_k) + \log p(w_{ik}) \end{aligned} \tag{18}$$

$\log p(m_k)$ and $\log p(w_{ik})$ can leave the expectation and cancel as they do not include any latent variables. Since KL-Divergence is always positive, and the function we are minimizing is the KL-Divergence between the variational family and the posterior, we can write the following inequality:

$$E_{q_{ik}}\Big[\log q_\phi(z_{ik}|m_k, w_{ik}, \boldsymbol{c_{ik}}) - \log p_\theta(z_{ik}|w_{ik}, m_k) - \sum_{j=1}^{2S} \log p_\theta(c_{ijk}|z_{ik})\Big] + \log p(\boldsymbol{c_{ik}}|m_k, w_{ik}) \geq 0 \tag{19}$$

Pushing the observed variables to the right-hand side of the inequality and negating both sides yields

$$E_{q_{ik}}\Big[-\log q_\phi(z_{ik}|m_k, w_{ik}, \boldsymbol{c_{ik}}) + \log p_\theta(z_{ik}|w_{ik}, m_k) + \sum_{j=1}^{2S} \log p_\theta(c_{ijk}|z_{ik})\Big] \leq \log p(\boldsymbol{c_{ik}}|m_k, w_{ik}) \tag{20}$$

To construct a lower-bound for the likelihood of context words given center word and metadata, $p(\boldsymbol{c_{ik}}|m_k, w_{ik})$, we minimize the negative left-hand side of Equation 20. That is, we minimize:

$$E_{q_{ik}}\Big[\log q_\phi(z_{ik}|m_k, w_{ik}, \boldsymbol{c_{ik}}) - \log p_\theta(z_{ik}|w_{ik}, m_k)\Big] - E_{q_{ik}}\Big[\sum_{j=1}^{2S} \log p_\theta(c_{ijk}|z_{ik})\Big] \tag{21}$$

We can write $E_{q_{ik}}\Big[\log q_\phi(z_{ik}|m_k, w_{ik}, \boldsymbol{c_{ik}}) - \log p_\theta(z_{ik}|w_{ik}, m_k)\Big]$ as the KL-Divergence between $q_\phi(z_{ik}|m_k, w_{ik}, \boldsymbol{c_{ik}})$ and $p_\theta(z_{ik}|w_{ik}, m_k)$. That is,

$$D_{KL}\Big(q_\phi(z_{ik}|m_k, w_{ik}, \boldsymbol{c_{ik}})||p_\theta(z_{ik}|w_{ik}, m_k)\Big) - E_{q_{ik}}\Big[\sum_{j=1}^{2S} \log p_\theta(c_{ijk}|z_{ik})\Big] \tag{22}$$

Using the definition of $p(c_{ijk}|z_{ik})$ and re-arranging terms,

$$D_{KL}\Big(q_\phi(z_{ik}|m_k, w_{ik}, \boldsymbol{c_{ik}})||p_\theta(z_{ik}|w_{ik}, m_k)\Big) \\ - \sum_{j=1}^{2S} E_{q_{ik}}\Big[\log \sum_m p_\theta(z_{ik}|c_{ijk}, m)p(m|c_{ijk})p(c_{ijk})\Big] \\ + E_{q_{ik}}\Big[\log E_{\tilde{c}}\Big[\sum_m p_\theta(z_{ik}|\tilde{c}, m)p(m|\tilde{c})\Big]\Big] \quad (23)$$

Here, we re-write $\sum_c \sum_d p_\theta(z_{ik}|c, d)p(d|c)p(c)$ in expected value form as $E_{\tilde{c}}\Big[\sum_d p_\theta(z_{ik}|\tilde{c}, d)p(d|\tilde{c})\Big]$. In addition, $p(c_{ijk})$ is the empirical probability value which does not contain the latent variable z_{ik}. Therefore, it can leave the expectation and be ignored during optimization:

$$D_{KL}\Big(q_\phi(z_{ik}|m_k, w_{ik}, \boldsymbol{c_{ik}})||p_\theta(z_{ik}|w_{ik}, m_k)\Big) \\ - \sum_{j=1}^{2S} E_{q_{ik}}\Big[\log \sum_m p_\theta(z_{ik}|c_{ijk}, m)p(m|c_{ijk})\Big] \\ + E_{q_{ik}}\Big[\log E_{\tilde{c}}\Big[\sum_m p_\theta(z_{ik}|\tilde{c}, m)p(m|\tilde{c})\Big]\Big] \quad (24)$$

Adding-subtracting $E_{q_{ik}}[\log q_\phi(z_{ik}|m_k, w_{ik}, \boldsymbol{c_{ik}})]$ to Equation 24 yields

$$D_{KL}\Big(q_\phi(z_{ik}|m_k, w_{ik}, \boldsymbol{c_{ik}})||p_\theta(z_{ik}|w_{ik}, m_k)\Big) \\ + \sum_{j=1}^{2S} E_{q_{ik}}\Big[\log q_\phi(z_{ik}|m_k, w_{ik}, \boldsymbol{c_{ik}}) - \log \sum_m p_\theta(z_{ik}|c_{ijk}, m)p(m|c_{ijk})\Big] \\ - E_{q_{ik}}\Big[\log q_\phi(z_{ik}|m_k, w_{ik}, \boldsymbol{c_{ik}}) - \log E_{\tilde{c}}\Big[\sum_m p_\theta(z_{ik}|\tilde{c}, m)p(m|\tilde{c})\Big]\Big] \quad (25)$$

This additional operation defines two KL-Divergence terms:

$$D_{KL}\Big(q_\phi(z_{ik}|m_k, w_{ik}, \boldsymbol{c_{ik}})||p_\theta(z_{ik}|w_{ik}, m_k)\Big) \\ + \sum_{j=1}^{2S} D_{KL}\Big(q_\phi(z_{ik}|m_k, w_{ik}, \boldsymbol{c_{ik}})|| \sum_m p_\theta(z_{ik}|c_{ijk}, d)p(m|c_{ijk})\Big) \\ - D_{KL}\Big(q_\phi(z_{ik}|m_k, w_{ik}, \boldsymbol{c_{ik}})||E_{\tilde{c}}\Big[\sum_m p_\theta(z_{ik}|\tilde{c}, m)p(m|\tilde{c})\Big]\Big) \quad (26)$$

To approximate $E_{\tilde{c}}\Big[\sum_m p_\theta(z_{ik}|\tilde{c}, m)p(m|\tilde{c})\Big]$, we sample a word using the negative word distribution (as in word2vec). As in the BSG model, we transform the second term into a hard margin to bound the loss in case the KL-Divergence terms for negatively sampled words are very large. The final objective we minimize is:

$$D_{KL}\Big(q_{ik}||p_{\theta}(z_{ik}|m_k, w_{ik})\Big)+$$
$$\sum_{j=1}^{2S}\max\Bigg(0, D_{KL}\Big(q_{ik}||\sum_{m}p_{\theta}(z_{ik}|c_{ijk}, m)\beta_{m|c_{ijk}}\Big) - D_{KL}\Big(q_{ik}||\sum_{m}p_{\theta}(z_{ik}|\tilde{c}, m)\beta_{m|\tilde{c}}\Big)\Bigg) \quad (27)$$

Here, we denote $q_{\phi}(z_{ik}|m_k, w_{ik}, \boldsymbol{c_{ik}})$ as q_{ik}. $\tilde{c}$ is sampled from $p(c)$ to construct an unbiased estimate for $E_{\tilde{c}}\Big[\sum_m p_{\theta}(z_{ik}|\tilde{c}, m)\beta_{m|\tilde{c}}\Big]$.

A.4. `variational network` Architecture

Words (w_{ik}, $\boldsymbol{c_{ik}}$), as well as metadata (m_k), are first projected onto a higher dimension via an embedding matrix E. The central word embedding $E_{w_{ik}}$ is then tiled across each context word and concatenated with context word embeddings $E_{\boldsymbol{c_{ik}}}$. We then encode the combined word sequence:

$$\boldsymbol{h} = LSTM(\big\{E_{\boldsymbol{c_{ik}}}; E_{w_{ik}}\big\}) \quad (28)$$

where ';' denotes concatenation and $\boldsymbol{h}$ represents the concatenation of the hidden states from the forward and backward passes at each timestep. The relevance of a word, especially one with multiple meanings, might depend on the section or document type in which it is found. To allow for an adaptive notion of relevance, we employ scaled dot-product attention (Vaswani et al., 2017) to compute a weighted-average summary of $\boldsymbol{h}$:

$$h_{word} = softmax(\frac{E_{m_k}^T\boldsymbol{h}}{\sqrt{dim_e}})\boldsymbol{h} \quad (29)$$

where dim_e is the embedding dimension. The scaling factor $\frac{1}{\sqrt{dim_e}}$ acts as a normalizer to the dot product. We selectively combine information from the metadata embedding (E_{m_k}) and attended context (h_{word}) with a gating mechanism similar to (Miyamoto and Cho, 2016). Precisely, we learn a relative weight[5]:

$$p_{m_k} = sigmoid(W_{gate}([E_{m_k}; h_{word}] + b_{gate})) \quad (30)$$

We then use p_{m_k} to create a weighted average:

$$h_{joint} = p_{m_k}E_{m_k} + (1 - p_{m_k})h_{word} \quad (31)$$

Finally, we project h_{joint} to produce isotropic Gaussian parameters

$$\mu_q = W_{\mu}h_{joint} + b_{\mu} \qquad \sigma_q = exp(W_{\sigma}h_{joint} + b_{\sigma}) \quad (32)$$

As in the BSG model, the network produces the log of the variance, which we exponentiate to ensure it is positive. We experimented with modeling a full covariance matrix. Yet, it did not improve performance and added immense cost to the KLD calculation.

[5] In practice, we compute separate relevance scores for word and metadata and apply the Tanh function before taking the softmax. We do this to place a constant lower bound on $min(p_{m_k}, 1 - p_{m_k})$ and prevent over-reliance on one form of evidence.

A.5. Additional Details on Experimental Setup

We provide explanations on a few key design choices for the experimental setup.

- **MIMIC RS Leakage:** It is important to note that we pre-train all models on the same set of documents which are used to create the synthetic MIMIC RS test set. While no acronym labels are provided during pre-training, we want to measure, and control for, any train-test leakage that may bias the reporting of the MIMIC RS results. Yet, we found removing all documents in the test set from pre-training degraded performance no more than one percentage point evenly across all models. For consistency and computational simplicity, we show performance for models pre-trained on all notes.

- **Mapping section headers from MIMIC to CASI and CUIMC**: We manually map sections in CASI and CUIMC for which no exact match exists in CUIMC. This is relatively infrequent, and we relied on simple intuition for most mappings. For example, one such transformation is *Chief Complaint* → *Chief Complaints.*

- **Choice of MLE over MAP estimate for section header baseline**: We choose the MLE over MAP estimate because the latter never selects rare LFs due to the huge class imbalances. This causes macro F1 scores to be very low.

- **LF phrases:** When an LF is a phrase, we take the mean of individual word embeddings.

A.5.1. Preprocessing

Clinical text is tokenized, stopwords are removed, and digits are standardized to a common format using the NLTK toolkit (Loper and Bird, 2002). The vocabulary comprises all terms with corpus frequency above 10. We use negative sampling with standard parameter 0.001 to downsample frequent words (Mikolov et al., 2013b). After preprocessing, the MIMIC pre-training dataset consists of $\sim 330m$ tokens, a token vocabulary size of $\sim 100k$, and a section vocabulary size of $\sim 10k$. We write a custom regex to extract section headers from MIMIC notes:

```
r'(?:^|\s{4,}|\n)[\d.#]{0,4}\s*([A-Z][A-z0-9/ ]+[A-z]:)'
```

The search targets a flexible combination of uppercase letters, beginning of line characters, and either a trailing ':' or sufficient space following a candidate header. We experimented with using template regexes to canonicalize section headers as well as concatenate note type with section headers. This additional hand-crafted complexity did not improve performance so we use the simpler solution for all experiments. The code exists to play around with more sophisticated extraction schemes.

A.5.2. Constructing CASI Test Set

For clarity into the results, we outline the filtering operations performed on the CASI dataset. In Table 6, we enumerate the operations and their associated reductions to the size of the original dataset. The final dataset at the bottom produces the gold standard test set against which all our models are evaluated. These changes were made in the interest of producing a coherent test set. Empirically, performance is not affected by the filtering operations.

Table 6: Filtering CASI Dataset.

Preprocessing Step	Examples
Initial	37,000
LF Same as SF (just a sense)	5,601
SF Not Present in Context	1,249
Parsing Issue	725
Duplicate Example	731
Single Target	1,481
SFs with LFs not present in MIMIC-III	8,976
Final Dataset	**18,233**

Because our evaluations rely on computing the distance between contextualized SFs and candidate LFs, we manually curate canonical forms for each LF in the CASI sense inventory. For instance, we replace the candidate LF for the acronym CVS:

$$\text{"customer, value, service"} \rightarrow \text{"CVS pharmacy;brand;store"}$$

where ';' represents a boolean *or*.

A.5.3. Hyperparameters

Our hyperparameter settings are shared across the LMC model and BSG baselines. We assign embedding dimensions of $100d$, and set all hidden state dimensions to $64d$. We apply a dropout rate of 0.2 consistently across neural layers (Srivastava et al., 2014). We use a hard margin of 1 for the hinge loss. Context window sizes are fixed to a minimum of 10 tokens and the nearest section/document boundary. We develop the model in PyTorch (Paszke et al., 2017) and train all models for 5 epochs with Adam (Kingma and Ba, 2014) for adaptive optimization (learning rate of $1e-3$). Inspired by denoising autoencoders (Vincent et al., 2008) and BERT, we randomly mask context tokens and central words with a probability of 0.2 during training for regularization. The conditional model probabilities $p(w|d)$ and $p(d|w)$ are computed with add-1 smoothing on corpus counts.

A.5.4. MBSGE Algorithm

The training procedure for MBSGE is enumerated in Algorithm 2, where m_k^1 represents the note type for the k'th document and m_{ik}^2 represents the section header corresponding to the i'th word in the k'th document. Rather than train three separate models, we train a single model with stochastic replacement to ensure a common embedding space. We choose non-uniform replacement sampling to account for the vastly different vocabulary sizes.

For evaluation, we average ensemble the Gaussian parameters from the `variational network` (q_ϕ), where x separately stands for both the center word acronym (w_{ik}), and the section header metadata (m_{ik}^2).

Algorithm 2 MBSGE Stochastic Training Procedure

```
while not converged do
    Sample $\boldsymbol{m_k}, w_{ik}, \boldsymbol{c_{ik}} \sim D$
    Sample $x \sim \text{Cat}(\{w_{ik}, m_k^1, m_{ik}^2\}; \{0.7, 0.1, 0.2\})$
    $\delta \leftarrow \nabla D_{KL}\Big(q_\phi(z_{ik}|x, \boldsymbol{c_{ik}})||p_\theta(z_{ik}|x)\Big)$
    $\phi, \theta \leftarrow$ Update parameters using $\delta$
```

Table 7: Aggregated across 5 pre-training runs. NLL is neg log likelihood, W/M weighted/macro.

		MIMIC				CUIMC				CASI			
	Model	NLL	Acc	W F1	M F1	NLL	Acc	W F1	M F1	NLL	Acc	W F1	M F1
Worst	BERT	1.36	0.40	0.40	0.33	1.41	0.37	0.33	0.28	1.23	0.42	0.38	0.23
	ELMo	1.34	0.56	0.59	0.51	1.39	0.55	0.58	0.48	1.21	0.51	0.52	0.36
	BSG	2.06	0.43	0.42	0.38	12.2	0.48	0.48	0.36	1.38	0.58	0.56	0.33
	MBSGE	1.26	0.60	0.62	0.54	7.94	0.61	0.61	0.48	0.96	0.68	0.67	0.43
	LMC	**0.82**	**0.74**	**0.77**	**0.68**	**0.91**	**0.69**	**0.68**	**0.56**	**0.80**	**0.71**	**0.73**	**0.50**
Mean	BERT	1.36	0.40	0.40	0.33	1.41	0.37	0.33	0.28	1.23	0.42	0.38	0.23
	ELMo	1.33	0.58	0.61	0.53	1.38	0.58	0.60	0.49	1.21	0.55	0.56	0.38
	BSG	1.28	0.57	0.59	0.52	9.04	0.58	0.58	0.46	0.99	0.64	0.64	0.41
	MBSGE	1.07	0.65	0.67	0.59	6.16	0.64	0.64	0.52	0.88	0.70	0.70	0.46
	LMC	**0.81**	**0.74**	**0.78**	**0.69**	**0.90**	**0.69**	**0.68**	**0.57**	**0.79**	**0.71**	**0.73**	**0.51**
Best	BERT	1.36	0.40	0.40	0.33	1.41	0.37	0.33	0.28	1.23	0.42	0.38	0.23
	ELMo	1.33	0.61	0.65	0.58	1.38	0.62	0.64	0.50	1.21	0.59	0.60	0.42
	BSG	0.98	0.64	0.68	0.59	5.41	0.61	0.62	0.50	0.85	0.67	0.70	0.46
	MBSGE	0.96	0.68	0.71	0.62	4.81	0.67	0.67	0.57	0.83	**0.72**	0.73	0.50
	LMC	**0.80**	**0.75**	**0.79**	**0.70**	**0.89**	**0.70**	**0.69**	**0.58**	**0.78**	**0.72**	**0.74**	**0.52**

A.6. Additional Evaluations

A.6.1. Aggregate Performance

In the main manuscript, we report mean results across the 5 pre-training runs. In Table 7, we include the best and worst performing models to provide a better sense of pre-training variance. Even though it is a small sample size, it appears the LMC is robust to randomness in weight initialization as evidenced by the tight bounds.

A.6.2. Bootstrapping

For robustness, we select the best performing from each model class and bootstrap the test set to construct confidence intervals. We draw 100 independent random samples from the test set and compute metrics for each model class. Each subset represents 80% of the original dataset. Very tight bounds exist for each model class as can be seen in Figure 6.

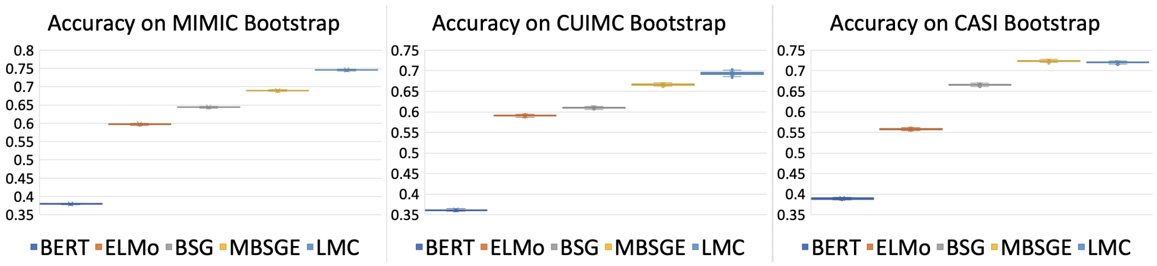


Figure 6: Confidence Intervals for Best Performing Models.

A.6.3. Effect of Number of Target Expansions

For most tasks, performance deteriorates as the number of target outputs grows. To measure the relative rate of decline, in Figure 7, we plot the F1 score as the number of candidate LFs increases.

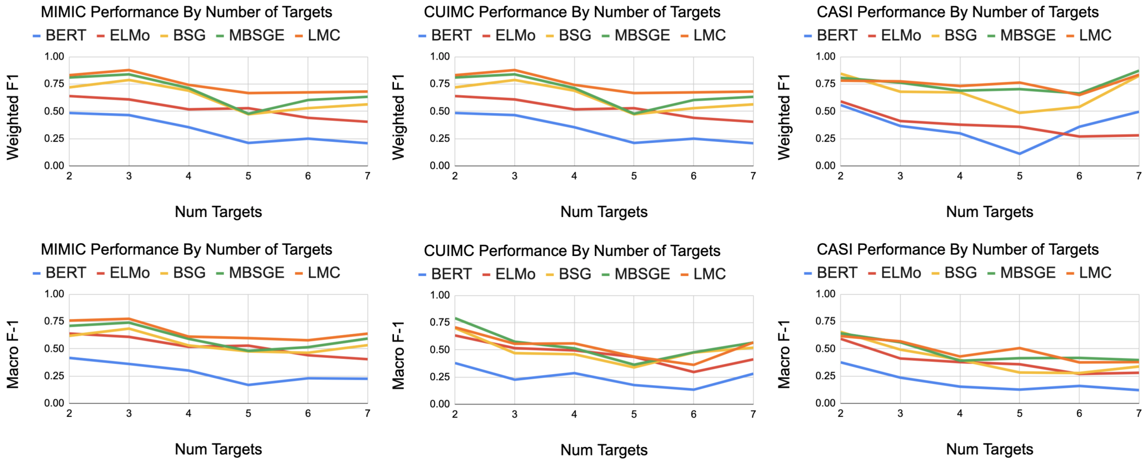


Figure 7: Effect of Number of Output Classes on F1 Performance. Best performing models shown.

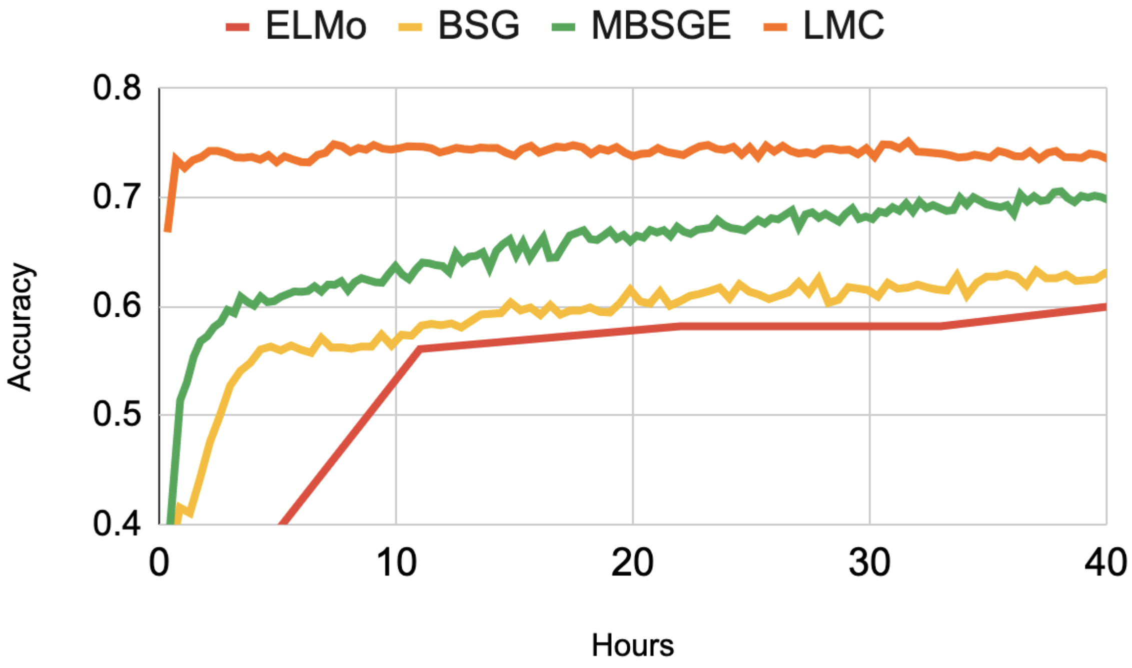
A.6.4. Acronym-Level Performance Breakdowns

We provide a breakdown of performance by SF on MIMIC RS between the LMC model and the ELMo baseline. There is a good deal of volatility across SFs, particularly for the macro F1 metric. We leave out the other baselines for space considerations.

			LMC						**ELMo**					
Acronym	Count	Targets	mPr	mR	mF1	wPr	wR	wF1	mPr	mR	mF1	wPr	wR	wF1
AMA	471	3	0.65	0.77	0.68	0.94	0.89	0.91	0.84	0.72	0.74	0.95	0.94	0.94
ASA	395	2	0.5	0.5	0.5	0.98	0.99	0.99	0.5	0.5	0.5	0.98	0.99	0.99
AV	491	3	0.57	0.69	0.58	0.88	0.79	0.82	0.58	0.41	0.13	0.92	0.08	0.11
BAL	485	2	0.68	0.84	0.72	0.93	0.87	0.89	0.64	0.87	0.65	0.93	0.78	0.83
BM	488	3	0.71	0.67	0.52	0.95	0.73	0.8	0.84	0.52	0.55	0.93	0.93	0.92
CnS	432	5	0.53	0.67	0.56	0.96	0.96	0.96	0.63	0.67	0.41	0.99	0.18	0.26
CEA	497	4	0.31	0.28	0.2	0.92	0.34	0.43	0.45	0.35	0.16	0.97	0.18	0.3
CR	499	6	0.47	0.61	0.38	0.97	0.84	0.88	0.17	0.17	0.01	0.91	0.04	0.01
CTA	495	4	0.49	0.44	0.46	0.98	0.94	0.96	0.51	0.89	0.49	0.97	0.85	0.91
CVA	474	2	0.93	0.91	0.91	0.92	0.92	0.91	0.78	0.5	0.37	0.76	0.57	0.42
CVP	487	3	0.61	0.76	0.51	0.92	0.63	0.75	0.45	0.56	0.44	0.91	0.72	0.77
CVS	237	3	0.47	0.78	0.38	0.88	0.47	0.53	0.47	0.34	0.34	0.78	0.78	0.74
DC	455	5	0.53	0.72	0.51	0.74	0.55	0.62	0.17	0.29	0.2	0.43	0.56	0.48
DIP	492	3	0.85	0.97	0.89	0.97	0.94	0.95	0.37	0.42	0.2	0.93	0.33	0.41
DM	484	3	0.61	0.86	0.57	0.92	0.78	0.83	0.65	0.75	0.51	0.97	0.64	0.77
DT	475	6	0.35	0.28	0.31	0.68	0.49	0.57	0.11	0.55	0.12	0.14	0.16	0.15
EC	473	4	0.59	0.74	0.54	0.95	0.93	0.93	0.27	0.59	0.16	0.1	0.04	0.05
ER	495	3	0.67	0.72	0.68	0.93	0.89	0.91	0.35	0.34	0.03	0.9	0.05	0.02
FSH	265	2	0.75	0.66	0.7	0.99	0.99	0.99	0.49	0.5	0.5	0.98	0.99	0.98
IA	171	2	0.51	0.74	0.35	0.99	0.49	0.64	0.51	0.5	0.02	0.99	0.02	0.01
IM	492	2	0.66	0.9	0.7	0.95	0.84	0.88	0.54	0.54	0.16	0.93	0.16	0.16
LA	454	3	0.7	0.98	0.75	0.99	0.98	0.99	0.48	0.62	0.19	0.96	0.06	0.05
LE	481	7	0.39	0.49	0.38	0.93	0.78	0.84	0.28	0.56	0.26	0.78	0.42	0.53
MR	492	5	0.44	0.62	0.35	0.96	0.5	0.63	0.42	0.72	0.26	0.92	0.34	0.31
MS	488	6	0.48	0.6	0.33	0.92	0.33	0.46	0.41	0.55	0.31	0.75	0.42	0.37
NAD	465	2	0.4	0.5	0.44	0.64	0.8	0.71	0.58	0.54	0.54	0.72	0.77	0.73
NP	463	4	0.44	0.58	0.48	0.93	0.87	0.89	0.53	0.38	0.32	0.91	0.88	0.84
OP	489	6	0.59	0.57	0.57	0.91	0.91	0.9	0.52	0.66	0.57	0.78	0.85	0.81
PA	412	6	0.38	0.48	0.29	0.82	0.46	0.44	0.43	0.43	0.26	0.92	0.35	0.36
PCP	488	4	0.44	0.59	0.32	0.67	0.43	0.45	0.48	0.41	0.35	0.77	0.44	0.45
PDA	478	3	0.49	0.53	0.48	0.83	0.74	0.75	0.86	0.8	0.82	0.8	0.81	0.79
PM	375	3	0.4	0.38	0.21	0.83	0.32	0.28	0.46	0.4	0.41	0.78	0.81	0.78
PR	241	4	0.67	0.75	0.58	0.82	0.71	0.76	0.6	0.35	0.31	0.78	0.47	0.39
PT	496	4	0.53	0.69	0.58	0.96	0.93	0.94	0.42	0.35	0.17	0.94	0.11	0.13
RA	490	4	0.5	0.57	0.47	0.91	0.72	0.78	0.66	0.56	0.58	0.91	0.9	0.9
RT	470	4	0.55	0.47	0.41	0.91	0.66	0.69	0.55	0.46	0.37	0.82	0.66	0.58
SA	454	5	0.8	0.77	0.61	0.99	0.84	0.85	0.58	0.65	0.48	0.73	0.82	0.77
SBP	489	2	0.59	0.55	0.24	0.86	0.25	0.2	0.64	0.74	0.54	0.87	0.57	0.61
US	290	2	0.91	0.91	0.91	0.92	0.92	0.92	0.86	0.61	0.6	0.82	0.75	0.69
VAD	482	4	0.44	0.48	0.25	0.9	0.37	0.51	0.25	0.27	0.04	0.8	0.07	0.13
VBG	483	2	0.79	0.75	0.7	0.83	0.71	0.7	0.96	0.95	0.95	0.96	0.95	0.95
AVG	-	-	**0.57**	**0.65**	**0.51**	**0.9**	**0.72**	**0.75**	**0.52**	**0.54**	**0.37**	**0.83**	**0.52**	**0.52**

A.7. Efficiency

Figure 8: Accuracy by pre-training hours. All plots flatten after 40 hours (not shown).



Task performance at the end of pre-training is an informative, but potentially incomplete, evaluation metric. Recent work has noted that large-scale transfer learning can come at a notable financial and environmental cost (Strubell et al., 2019). Also, a model which adapts quickly to a task may emulate general linguistic intelligence (Yogatama et al., 2019). In Figure A.7, we plot test set accuracy on MIMIC RS at successive pre-training checkpoints. We pre-train the models on a single NVIDIA GeForce RTX 2080 Ti GPU. We hypothesize that flexibility in latent word senses and shared statistical strength across section headers facilitate rapid LMC convergence. Averaged across datasets and runs, the number of pre-training hours required for peak test set performance is 6 for LMC, while 50, 51, and 55 for MBSGE, BSG, and ELMo. The non-embedding parameter counts are 169k for the LMC and 150k for both the BSG and MBSGE. ELMo has 91mn parameters. Taken together, the LMC efficiently learns the task as a by-product of pre-training.

A.8. Words and Metadata as Mixtures

Consider metadata and its building blocks. A natural question to consider is the distribution of latent meanings given metadata. We can simply write this as

$$p(z_{ik}|m_k) = \sum_{w_{ik}} p(z_{ik}|w_{ik}, m_k)p(w_{ik}|m_k) \tag{33}$$

w_{ik} denotes an arbitrary word in document k and the summation marginalizes it with respect to the vocabulary. $p(w_{ik}|m_k)$ can be measured empirically with corpus statistics. We will denote this probability value as $\xi_{w_{ik}|m_k}$. In addition, $p(z_{ik}|, w_{ik}, m_k)$ has already been defined as $N(nn(w_{ik}, m_k; \theta))$. Therefore,

$$p(z_{ik}|m_k) = \sum_{w_{ik}} N(nn(w_{ik}, m_k; \theta))\xi_{w_{ik}|m_k} \tag{34}$$

The distribution of the latent space over metadata is a mixture of Gaussians weighted by occurrence probability in metadata k. One can measure the similarity between two metadata using KL-Divergence. This measure is computationally expensive because each metadata can be a mixture of thousands of Gaussians. Monte Carlo sampling, however, can serve as an efficient, unbiased approximation.

It is also a natural question to ask about the potential meanings a word can exhibit (Figure 9). That is,

$$p(z_{ik}|w_{ik}) = \sum_{m_k} p(z|m_k, w_{ik})p(m_k|w_{ik}) \tag{35}$$

$p(m_k|w_{ik})$ can also be measured empirically. We denote this distribution as $\beta_{m_k|w_{ik}}$.

$$p(z_{ik}|w_{ik}) = \sum_{m_k} N(nn(w_{ik}, m_k; \theta))\beta_{m_k|w_{ik}} \tag{36}$$

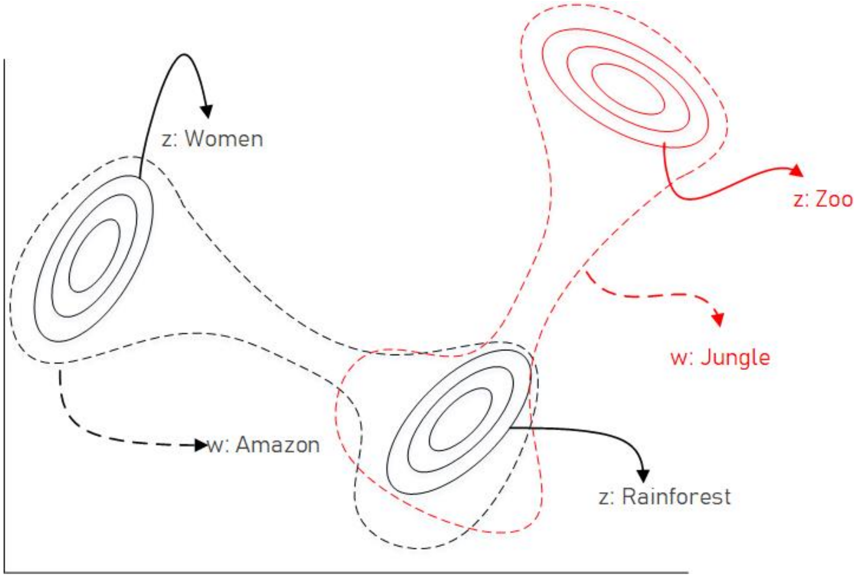


Figure 9: The meaning of "Amazon" can be interpreted as a mixture of Gaussian distributions in different metadata.

A.9. Word and Metadata as Vectors

With a certain trade-off of compression, we can represent metadata as a vector using its expected conditional meaning:

$$E_{z_{ik}|m_k}[z_{ik}] = \sum_{w_{ik}} \xi_{w_{ik}|m_k} \int z_{ik} N(nn(w_{ik}, m_k; \theta)) dz_{ik} \tag{37}$$

Since $\int z_{ik} N(nn(w_{ik}, m_k; \theta)) dz_{ik} = E_{z_{ik}|w_{ik},m_k}[z_{ik}]$ The expectation can be simply written as the combination of the means of normal distributions that form metadata k:

$$E_{z_{ik}|m_k}[z_{ik}] = \sum_{w_{ik}} \xi_{w_{ik}|m_k} E[z_{ik}|w_{ik}, m_k] \tag{38}$$

The above equation sums the expected meaning of words inside a metadata weighted by occurrence probability. Following the same logic for words yields

$$E_{z_{ik}|w_{ik}}[z_{ik}] = \sum_{m_k} \beta_{m_k|w_{ik}} E[z_{ik}|w_{ik}, m_k] \tag{39}$$