Proceedings of Machine Learning Research 174:234–247, 2022 Conference on Health, Inference, and Learning (CHIL) 2022

# Context-Sensitive Spelling Correction of Clinical Text via Conditional Independence

**Juyong Kim** JUYONGK@CS.CMU.EDU
*Machine Learning Department, Carnegie Mellon University, Pittsburgh, PA 15213*

**Jeremy C. Weiss** JEREMYWEISS@CMU.EDU
*Heinz College of Information Systems and Public Policy, Carnegie Mellon University, Pittsburgh, PA 15213*

**Pradeep Ravikumar** PRADEEPR@CS.CMU.EDU
*Machine Learning Department, Carnegie Mellon University, Pittsburgh, PA 15213*

## Abstract

Spelling correction is a particularly important problem in clinical natural language processing because of the abundant occurrence of misspellings in medical records. However, the scarcity of labeled datasets in a clinical context makes it hard to build a machine learning system for such clinical spelling correction. In this work, we present a probabilistic model of correcting misspellings based on a simple conditional independence assumption, which leads to a modular decomposition into a language model and a corruption model. With a deep character-level language model trained on a large clinical corpus, and a simple edit-based corruption model, we can build a spelling correction model with small or no real data. Experimental results show that our model significantly outperforms baselines on two healthcare spelling correction datasets.

**Data and Code Availability** This paper uses the MIMIC-III dataset (Johnson et al., 2016), which is available on the PhysioNet repository (Moody et al., 2000), and misspelling annotation of the MIMIC-III dataset by (Fivez et al., 2017). This paper also uses a dataset released by CSpell (Lu et al., 2019) which contains spelling errors in consumer health questions. For the reviewing purpose, we append the anonymized version of the code as supplementary material. The source code repository and the model parameter are available at `https://github.com/dalgu90/cim-misspelling`.

## 1. Introduction

Spelling correction is an old problem in natural language processing and is especially essential in healthcare environments that are rife with error-prone text. Estimates of spelling error rates in clinical notes range from 0.4% (Lai et al., 2015) to 7% (Tolentino et al., 2007). Correcting misspelled words in clinical texts is crucial since misspellings can have a notorious effect on downstream NLP tasks such as automatic diagnostic coding (assigning diagnosis codes given the clinical note).

Such spelling correction in a clinical context has several challenges. First, the candidate generation step, a critical component of most spelling correction methods, requires time complexity proportional to the vocabulary size. Second, the context representations, such as n-gram and word embeddings, of context-sensitive methods cannot properly handle rare words or words not in the embedding dictionary. In healthcare settings, both issues above are more severe than in general English text because of more extensive and specialized clinical vocabulary.

In this work, we propose a probabilistic model of correcting misspellings, named Conditional Independence Model (CIM), where we compute the posterior probability of the correct word given the misspelled word and the context. We assume that the misspelling is independent of the context given the correct word, and under this assumption, the posterior probability can be decomposed into a language model component and a corruption model component. Using a character-level language model and a simple edit-based corruption model, we can naturally decode the correct word using beam search.

This simple approach addresses both of the caveats raised earlier. CIM generates output candidates in a computationally inexpensive auto-regressive manner. And the deep language model provides the probability of any candidate given the context, which solves the issue of rare or out-of-embedding words. Moreover, our approach is modular, and allows for incorporating any advances in both language models, as well as edit-based corruption models. We validate the effectiveness of CIM with the two healthcare misspelling datasets and show CIM can be trained and tuned with fully unsupervised settings. To our knowledge, this work is the first approach to the spelling correction problem with the noisy channel model combined with a deep character-level language model.

**Example 1.**
○ **Input text:** "... with hazy ground-glass opacity in the lower lobes of the **lugns** . There is a small amount of perihepatic ascites. The evaluation of the abdomen ..."
○ **Correction**: "**lungs**"
○ **Vectors:**

- Misspelled word $\mathbf{y} = [\mathrm{l}, \mathrm{u}, \mathrm{g}, \mathrm{n}, \mathrm{s}]$
- Context $\mathbf{c} = [\mathbf{c}_{\text{left}}, \mathbf{c}_{\text{right}}]$
  - $\mathbf{c}_{\text{left}} = [..., \text{with}, \text{hazy}, ..., \text{lobes}, \text{of}, \text{the}]$
  - $\mathbf{c}_{\text{right}} = [\text{“.”}, \text{there}, \text{is}, ..., \text{the}, \text{abdomen}, ...]$
- Correction $\mathbf{x} = [\mathrm{l}, \mathrm{u}, \mathrm{n}, \mathrm{g}, \mathrm{s}]$

## 2. Related Works

**Spelling Error Correction** Spelling correction, a sub-problem within spell checking, is the problem of correcting a given misspelled word. One of the earliest attempts of spelling correction is based on edit distance (Damerau, 1964). A Bayesian approach to spelling correction is the noisy channel model (Kemighan et al., 1990; Brill and Moore, 2000), which computes the correction posterior given a word prior and a corruption model. As we detail later, our approach extends this to the more modern setting which includes word contexts.

In a more modern context, there have been several approaches to detect and correct misspellings with deep neural networks. Li et al. (2018) uses a nested RNN to encode input from character-level embeddings. Li et al. (2020) uses Transformer encoder at word- and character-level. Jayanthi et al. (2020) performed a comprehensive comparison among deep models on synthetic and real misspelling correction dataset.

Compared to these approaches, we have several advantages in correcting misspellings. First, our model adopts a character-level language model and easily generalizes to rare or even unseen words, which is highly advantageous in a clinical setting where the size of the vocabulary is large. Previous models output corrections by $|V|$-way multi-class classification, where $V$ is the vocabulary. Also, our approach requires a small labeled misspelling dataset only for tuning hyper-parameters of the corruption model. Previous approaches require a large number of labeled misspelling examples to train the classifier and resort to synthetically generated training data, which has risk of distribution mismatch from real misspellings.

In healthcare settings, there have been several works on developing misspelling detection and correction methods (Ruch et al., 2003; Tolentino et al., 2007; Lai et al., 2015). Fivez et al. (2017) develops a spelling correction algorithm using orthographic and phonetic edit distances and word embedding similarity. Lu et al. (2019) develops a pipeline that detects and corrects various types of spelling errors using simple rules and word embeddings. The two papers above also released real datasets of spelling errors to evaluate their performance.

**Contextualized Language Models** Transfer learning from pre-trained deep language models have revolutionized NLP in recent years, especially from the introduction of the Transformer architecture (Vaswani et al., 2017). BERT (Devlin et al., 2018) uses the Transformer encoder and solves auxiliary language tasks to pre-train word embeddings. GPT (Radford et al., 2018) adopts the Transformer decoder to generate languages in an autoregressive manner. Similar to BART (Lewis et al., 2020), our model uses both Transformer encoder and decoder to take the context as input and output the correction word.

## 3. Methods

In this section, we introduce mathematical notations, the spelling correction problem, and our proposed method, CIM.

### 3.1. Problem Setup

Let $W = \{w_1, \cdots, w_{|W|}\}$ be a set of characters. This includes characters and punctuation marks used in the language of interest. The misspelled word $\mathbf{y} = [y_1, ..., y_{N_Y}] \in W^{N_Y}$, and its correction $\mathbf{x} = [x_1, ..., x_{N_X}] \in W^{N_X}$ are both sequences of characters. This character level representation of words is more suitable than subwords in our problem since the

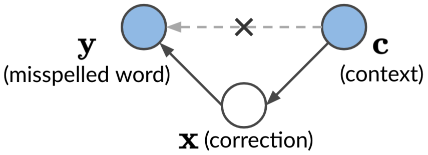


Figure 1: Graphical model of our conditional independence model. The context and the typo are observed and the correct word is unobserved.

typo and the correct words substantially differ when tokenized into subwords.

Next, we define the context. We denote the vocabulary, the set of subwords used by a tokenizer, as $V = \{v_1, \cdots, v_{|V|}\}$. The context $\mathbf{c} = [\mathbf{c}_{\text{left}}, \mathbf{c}_{\text{right}}] \in V^L$, where $\mathbf{c}_{\text{left}} = [c_{-L_{\text{left}}}, ..., c_{-1}]$ and $\mathbf{c}_{\text{right}} = [c_1, ..., c_{L_{\text{right}}}]$ are part of the text occurring before and after the typo word. The overall length of the context $L = L_{\text{left}} + L_{\text{right}}$ is typically constrained by language models such as BERT.

Spelling correction, the task of finding the correction given the misspelled word and the context, can be written as the following *probabilistic inference task*:

$$\mathbf{x} = \arg\max_{\mathbf{x} \in W^{\leq N}} p(\mathbf{x}|\mathbf{y}, \mathbf{c}),$$

where $N$ is the maximum length of the correction. An illustrated example of misspelling correction problem with the notation is shown in Example 1.

### 3.2. Conditional Independence Model (CIM)

This subsection describes our method of spelling correction. Here, we make an assumption on the generative process of the misspelling that it only depends on the correct word, not the context. Hence, the typo word is independent to the context given the correct word: $\mathbf{y} \perp\!\!\!\perp \mathbf{c} \,|\, \mathbf{x}$. This is a reasonable assumption given that the most cause of misspellings (homophones, typographical errors, mispronunciations) are independent to surrounding words. Please see Figure 1 for the model.

With this assumption, we can express the MAP estimator of the correction as follows:

$$\begin{aligned} \mathbf{x} &= \arg\max_{\mathbf{x}} p(\mathbf{x}|\mathbf{y}, \mathbf{c}) = \arg\max_{\mathbf{x}} p(\mathbf{x}, \mathbf{y}, \mathbf{c}) \\ &= \arg\max_{\mathbf{x}} p(\mathbf{c}) p(\mathbf{x}|\mathbf{c}) p(\mathbf{y}|\mathbf{x}) \\ &= \arg\max_{\mathbf{x}} \{\log p(\mathbf{x}|\mathbf{c}) + \log p(\mathbf{y}|\mathbf{x})\} \end{aligned}$$

Note that the proposed model is similar to the noisy channel model, but we include the word context which in turn entails our specific conditional independence assumption.

The first term is the language model which is the probability distribution of the correct word given the context. We model this as a transformer encoder-decoder architecture (Lewis et al., 2020), where the encoder is bidirectional same as BERT, and the decoder outputs sequence of characters in an auto-regressive, or left-to-right, manner.

The second term is the corruption model, the probability model of the typo word given the correct word. This may take into account delicate mechanism such as proximity in keyboard layout or be learned if a large dataset of misspelling is available. We adopt a simple approach that the probability is proportional to the exponential of the character edit distance between the correct word and typo word:

$$\log p(\mathbf{y}|\mathbf{x}) = -C d_{\text{ED}}(\mathbf{x}, \mathbf{y}),$$

where $d_{\text{ED}}(\cdot, \cdot)$ is the Damerau-Levenshtein edit distance, and $C$ is a hyper-parameter that balances between the language model and the corruption model. We chose a simple corruption model for two reasons: to demonstrate the efficacy of the overall approach even with a simple baseline, but also that it allows us to setup a correction system with little or no data, since more complex corruption models require a training dataset of "typical misspellings" which might not always be available.

Thus since the corruption model above does not require any training, our method only requires training the character-level language model of the correct word, which can be performed with a large clinical corpus and does not require a dataset of misspellings. We further note that our approach is modular, and allows for incorporating any future advances in either language models, or corruption models (for instance if large-scale misspellings datasets become available).

### 3.3. Beam Search with the Two Model Components

After training the language model, we combine the two models at decoding phase to perform misspelling correction. At the time step $t$ of beam search, the model outputs candidates by expanding candidates from the previous time step and sorting them by in-

○ **Input text:** "... opacity in the lower lobes of the **lugns** . There is a small amount of perihepatic ..."
○ **Misspelled word:** "**lugns**"

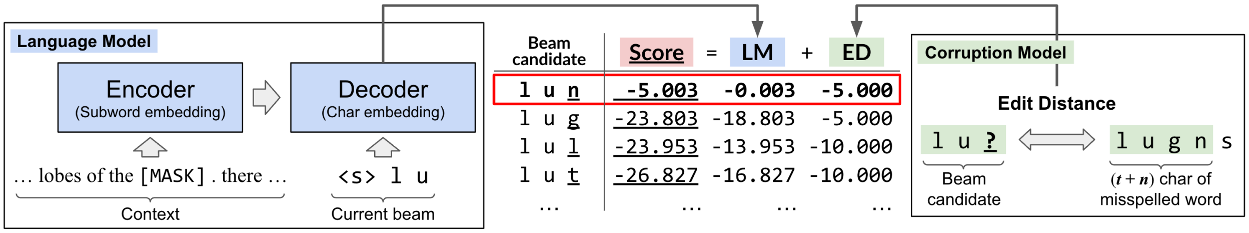

Language Model
Encoder
(Subword embedding)
Decoder
(Char embedding)
… lobes of the [MASK] . there …
Context
<s> l u
Current beam
Beam candidate
Score = LM + ED
l u n  -5.003  -0.003  -5.000
l u g  -23.803  -18.803  -5.000
l u l  -23.953  -13.953  -10.000
l u t  -26.827  -16.827  -10.000
…  …  …  …
Corruption Model
Edit Distance
l u ?
l u g n s
Beam candidate
(t + n) char of misspelled word

Figure 2: Beam search of CIM on the Example 1 at time step $t = 3$. The beam candidates are ranked by the sum of the language model score (LM) and the corruption model score (ED). The hyper-parameters of the corruption model are $C = 5.0$ and $n = 1$. The beam width is chosen to $B = 1$ for clear visualization.

termediate scores:

$$\mathcal{B}_t = \underset{\mathbf{x}_{:t} \in (\mathcal{B}_{t-1} \times W)}{\arg\max_B} \left\{ \log p(\mathbf{x}_{:t}|\mathbf{c}) + \log p(\mathbf{y}|\mathbf{x}_{:t}) \right\},$$

where $\mathcal{B}_t$ is the set of candidates up to length $t$, $B$ is the beam width, and $\arg\max_k$ is top-$k$ argmax.

The first term, the language model score can be obtained naturally. The second term, the corruption model score is computed by the edit distance of the partial output and the first $t + n$ characters of the typo word:

$$\log p(\mathbf{y}|\mathbf{x}_{:t}) = -C d_{\text{ED}}(\mathbf{x}_{:t}, \mathbf{y}_{:t+n}).$$

This prevents the edit distance score from advantaging the characters of the typo word far behind the $t$-th position. The hyper-parameter $n$ implicitly assumes how many characters can be inserted, at most, to corrupt a word. Also, we restrict the possible set of the correct words to be the predefined dictionary. Combining the two scores and the dictionary constraint, the beam search step of our method becomes as follows:

$$\mathcal{B}_t = \underset{\mathbf{x}_{:t} \in (\mathcal{B}_{t-1} \times W) \cap D_t}{\arg\max_B} \left\{ \sum_{i=1}^{t} \log p(\mathbf{x}_i | \mathbf{x}_{<i}, \mathbf{c}) - C d_{\text{ED}}(\mathbf{x}_{:t}, \mathbf{y}_{:t+n}) \right\},$$

where $D_t$ is the set of length-$t$ substrings of the dictionary words. After generating the candidate words, we choose the best candidate by the scores of the candidates normalized by their lengths, following the practice of beam search. Please see Figure 2 for the visualization of the beam search of CIM.

## 4. Experiments

### 4.1. Datasets

There are few publicly available clinical misspelling datasets annotated by human experts. Here we describe two datasets to tune and evaluate our method for misspelling correction. Please refer to Appendix A for detailed description of the data processing.

**MIMIC-III Misspelling Dataset** Fivez et al. (2017) released a manually annotated dataset of clinical misspellings from the clinical notes in the MIMIC-III database. This single-set dataset contains 873 instances of 357 non-word misspellings. After a carefully review of the examples by a medical doctor, we found that the labels of 30 examples are incorrect. We updated the labels of these examples when we evaluate our method and the misspelling correction method proposed in Fivez et al. (2017).

**CSpell Spelling Error Dataset** Lu et al. (2019) released a dataset of various types of spelling errors. The dataset is collected from consumer health questions to their QA system and covers a wide range of errors other than misspellings, such as to-merge and to-split errors. Since we focus on evaluating misspelling correction, we extract single-word misspellings that contain only alphabets. Note that this dataset is harder than the MIMIC-III dataset since it contains both real-word and non-word misspellings, and the questions are written in an informal language.

Their spelling checking software, CSpell, both detects and corrects spelling errors. To make a fair comparison with CSpell, we further excluded examples from the test set that are not detected by any of the detection modules of CSpell. As results, 409 and 574 examples are chosen from the training set and the test set, respectively.

### 4.2. Implementation

The implementation of the language model of CIM is based on the BART implementation of Hugging Face's Transformers (Wolf et al., 2020). The encoder part is same as BERT (Devlin et al., 2018) and initialized with BlueBERT (Peng et al., 2019), a clinical version of BERT. We use the same number of Transformer decoder layers as the encoder. We denote the model with BlueBERT-Large (24-layer) as CIM-Large and the model with BlueBERT-Base (12-layer) as CIM-Base. As a result, CIM-Base and CIM-Large have 132M and 403M parameters, respectively.

The reference dictionary is built by combining an English dictionary DWYL (2020) and a medical lexicon, the `LRWD` and `prevariants` tables of Unified Medical Language System (UMLS). The dictionary is used to select a valid word during the training of the language model and to restrict the possible beam candidates during the decoding. For computational efficiency, a Trie, or a prefix tree, is built to get the possible beam candidates.

### 4.3. Training and Evaluation

We trained the language model of CIM on the clinical notes of the MIMIC-III dataset. Both CIM-Base and CIM-Large are trained for 500k iterations with batch size 256 on 4×NVIDIA A100 GPU 40GB. We follow the optimizer and learning rate schedule of BERT, except we reduced the learning rate of the encoder since the encoder is initialized with BlueBERT while the decoder is randomly initialized. Hyper-parameter search is performed over various values of $C$, $n$, and the training steps to maximize the correction accuracy on the CSpell training set. The beam width is fixed to $B$=30 during the tuning for faster search. Please refer to Appendix B for further details.

## 5. Results and Analysis

In this section, we report the results of CIM on the two real datasets of clinical misspelling, and perform additional analyses on CIM.

| **Method** | **Acc(%)** |
|---|---|
| Fivez et al. (2017) | 92.10 |
| CIM-Base ($B$=30) | 95.07 |
| CIM-Base ($B$=300) | 95.99 |
| CIM-Large ($B$=30) | 95.65 |
| CIM-Large ($B$=300) | **96.56** |

Table 1: Accuracy results for the MIMIC-III misspelling datasets ($B$ is the width of beam search).

| **Method** | **Acc(%)** |
|---|---|
| Lu et al. (2019) | 54.70 |
| CIM-Base ($B$=30) | 67.07 |
| CIM-Base ($B$=300) | **69.51** |
| CIM-Large ($B$=30) | 65.68 |
| CIM-Large ($B$=300) | 67.60 |

Table 2: Accuracy results for the CSpell test set ($B$ is the width of beam search).

| **Hyper-param** | **Models** | |
|---|---|---|
| | CIM-Base | CIM-Large |
| Early stopping | 475k steps | 300k steps |
| $C$ | 5.0 | 5.0 |
| $n$ | 1 | 1 |

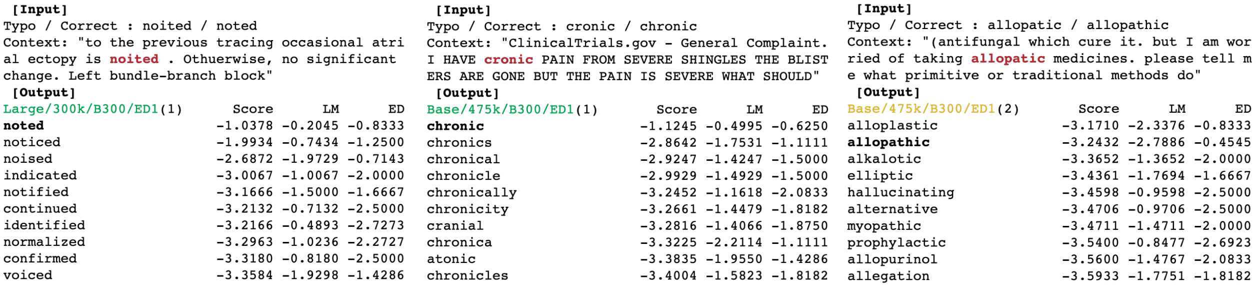
Table 3: Hyper-parameters of CIM tuned on CSpell training dataset.

### 5.1. Results on Clinical Misspelling Datasets

We report our results and those of baselines for the two real misspelling datasets in Table 1 and Table 2. The hyper-parameters chosen for Beam search decoding are shown in Table 3. For both of the datasets, our method outperforms the dataset baselines by large margin.

In all settings, the accuracy increases as the beam width increases. One interesting observation is that CIM-Large performs better than CIM-Base on the MIMIC-III dataset but worse on the CSpell dataset. We think that this is because CIM-Large overfits the word distribution of the MIMIC-III notes, which is different from the health consumer questions.

Figure 3 shows some beam search examples of CIM. The candidates are generated and ranked by their



```
 [Input]
Typo / Correct : noited / noted
Context: "to the previous tracing occasional atri
al ectopy is noited . Othuerwise, no significant
change. Left bundle-branch block"
 [Output]
Large/300k/B300/ED1(1)           Score      LM      ED
noted                          -1.0378 -0.2045 -0.8333
noticed                        -1.9934 -0.7434 -1.2500
noised                         -2.6872 -1.9729 -0.7143
indicated                      -3.0067 -1.0067 -2.0000
notified                       -3.1666 -1.5000 -1.6667
continued                      -3.2132 -0.7132 -2.5000
identified                     -3.2166 -0.4893 -2.7273
normalized                     -3.2963 -1.0236 -2.2727
confirmed                      -3.3180 -0.8180 -2.5000
voiced                         -3.3584 -1.9298 -1.4286
```

(*a*) MIMIC-III example (success)

```
 [Input]
Typo / Correct : cronic / chronic
Context: "ClinicalTrials.gov - General Complaint.
I HAVE cronic PAIN FROM SEVERE SHINGLES THE BLIST
ERS ARE GONE BUT THE PAIN IS SEVERE WHAT SHOULD"
 [Output]
Base/475k/B300/ED1(1)            Score      LM      ED
chronic                        -1.1245 -0.4995 -0.6250
chronics                       -2.8642 -1.7531 -1.1111
chronical                      -2.9247 -1.4247 -1.5000
chronicle                      -2.9929 -1.4929 -1.5000
chronically                    -3.2452 -1.1618 -2.0833
chronicity                     -3.2661 -1.4479 -1.8182
cranial                        -3.2816 -1.4066 -1.8750
chronica                       -3.3225 -2.2114 -1.1111
atonic                         -3.3835 -1.9550 -1.4286
chronicles                     -3.4004 -1.5823 -1.8182
```

(*b*) CSpell Test example (success)

```
 [Input]
Typo / Correct : allopatic / allopathic
Context: "(antifungal which cure it. but I am wor
ried of taking allopatic medicines. please tell m
e what primitive or traditional methods do"
 [Output]
Base/475k/B300/ED1(2)            Score      LM      ED
alloplastic                    -3.1710 -2.3376 -0.8333
allopathic                     -3.2432 -2.7886 -0.4545
alkalotic                      -3.3652 -1.3652 -2.0000
elliptic                       -3.4361 -1.7694 -1.6667
hallucinating                  -3.4598 -0.9598 -2.5000
alternative                    -3.4706 -0.9706 -2.5000
myopathic                      -3.4711 -1.4711 -2.0000
prophylactic                   -3.5400 -0.8477 -2.6923
allopurinol                    -3.5600 -1.4767 -2.0833
allegation                     -3.5933 -1.7751 -1.8182
```

(*c*) CSpell Test example (failure)

Figure 3: Beam search decoding examples. For each example, we display the top 10 beam candidates. The column next to the candidate (`Score`) shows the final beam score for each candidate.

scores (`Score`) consisting of the language model score (`LM`) and corruption model score (`ED`). Figure 3(*b*) shows an easy case where both modules give the highest scores to the correct word. In a more challenging example such as Figure 3(*a*), there is a candidate word ("noised") that has a higher corruption model score than the correct word ("noted"), but the language model gives a much higher score to the correct word. Figure 3(*c*) shows a failure case of CIM, where the language model gave a low score to the correct word. See Appendix E for more decoding results.

### 5.2. Analysis

**Ablation Study of Model Components** To see the effect of each component of our misspelling correction model, we conducted an ablation study of the corruption model and the reference dictionary. For each configuration, the hyper-parameters are tuned independently, and the evaluation was performed with the beam width $B = 300$.

The first four rows of Table 4 shows the results of the ablation study. The most noticeable result is that the corruption model contributes significantly to the model's performance. This is predictable because, with only the language model, the output would be any word that fits into the context, regardless of the misspelled word. Another observation is that dictionary matching contributes to the model only when both the language model and the corruption model are used. This is because the reference dictionary is unnecessary for the "LM only" setting since the language model is trained to produce dictionary words. However, when combined with the corruption model, the chance to output non-dictionary words increases, so the reference dictionary helps our model.

| **Method \ Dataset** | MIMIC-III | | CSpell Test | |
|---|---|---|---|---|
| | Base | Large | Base | Large |
| LM only | 37.57 | 37.34 | 20.91 | 20.21 |
| LM + Dict | 37.57 | 37.34 | 20.91 | 20.21 |
| LM + ED | 93.24 | 92.67 | 66.72 | 66.03 |
| LM + ED + Dict | **95.99** | **96.56** | **69.51** | 67.60 |
| Unsupervised | 95.07 | 95.42 | 68.29 | **68.99** |

Table 4: Accuracy results of ablation study and unsupervised setting. LM: language model, ED: corruption model, Dict: dictionary matching, Unsupervised: tuning on the synthetic dataset ($B$=300)

| $C \setminus n$ | 0 | 1 | 2 | $\infty$ |
|---|---|---|---|---|
| 2.0 | 64.29 | 64.63 | 63.94 | 61.32 |
| 5.0 | 66.03 | **67.07** | 66.20 | 63.94 |
| 10.0 | 56.97 | 58.01 | 57.32 | 52.96 |
| 20.0 | 46.34 | 49.83 | 48.78 | 44.25 |

Table 5: Accuracy results on the CSpell test set with different values of $C$ and $n$ (CIM-Base with $B$=30).

**Effect of Hyper-parameters** To see the effects of the hyper-parameters, we evaluate our model with various values of $C$ and $n$. Table 5 shows the results of CIM-Base on the CSpell test set with various $C$ and $n$. We choose $B = 30$ for the beam search. We can see that the best accuracy is achieved when the hyper-parameters are tuned to the CSpell train set ($C = 5.0$ and $n = 1$), which suggests the CSpell test distribution align with the train set. Also, the

| **Dataset \ Semantic Type** | | Substance | Disease | Symptom | Other | (Total) |
|---|---|---|---|---|---|---|
| **MIMIC-III** | Fivez et al. (2017) | 91.24 | 89.87 | 94.83 | 91.52 | 92.10 |
| | CIM | **95.62** | **96.20** | **98.28** | **96.34** | **96.56** |
| | Count | 137 | 79 | 174 | 500 | 873 |
| **CSpell Test** | Lu et al. (2019) | 54.31 | 70.59 | 57.89 | 48.77 | 54.70 |
| | CIM | **68.10** | **72.94** | **77.63** | **66.05** | **69.51** |
| | Count | 116 | 85 | 76 | 324 | 574 |

Table 6: Results of subgroup analysis by UMLS Semantic Types

accuracy decreases as the values of $C$ and $n$ move away from their optimal values. This is predictable since these hyper-parameters balances the model's preference for different candidates. In other words, increasing $C$ makes the model prefer candidates similar to the misspelled word, and decreasing it makes the model prefer candidates fitting the context better.

**Subgroup Analysis by Semantic Types** To see the effectiveness of CIM on different types of words, we computed the subgroup accuracy according to UMLS Semantic Type. We first chose three subtrees in the UMLS Semantic Type hierarchy for three subgroups of words, namely "substance", "disease", and "symptom". Then, for each example, we query the correction word to UMLS for semantic types and include the example into a subgroup if any semantic types of the correction word fall into the subgroup. We also grouped examples that did not belong to any of the subgroups into "other" subgroup. Note that an example can belong to more than one subgroup. Please refer to Appendix C for further details.

Table 6 shows the results of the subgroup analysis. In all subgroups of both datasets, CIM consistently outperforms baselines. In the CSpell dataset, CIM performs better on "Disease" and "symptom" subgroups than others, and similarly in the MIMIC-III test set, CIM shows the best accuracy on "Symptom" subgroup and most significant improvement over baseline on "Disease" subgroup.

**Results in Unsupervised Setting** While we used a small set of real misspellings for the hyper-parameter search, we can tune the hyper-parameters even in an unsupervised setting with a synthetic dataset of misspellings. From the MIMIC-III clinical notes, we randomly choose words that are in our reference dictionary and corrupt them with random operations, which results in 10k examples of synthetic misspellings. To corrupt words, up to two operations of character addition, deletion, substitution, or transposition can be applied.

We performed the hyper-parameter search on this synthetic dataset, as we did with the CSpell training set. The last row of Table 4 shows the results, and the test accuracy under the fully unsupervised setting is comparable to the supervised setting. Please refer to Appendix D for the data generation method and the resulting hyper-parameters.

## 6. Conclusion

The main contribution of the paper is to present a well-formalized spelling correction method that combines a deep neural language model and the corruption model. Our experiments show that the method outperforms the baseline methods, including an off-the-shelf software. Although the main concern of the paper is healthcare text, our method can be applied to other areas with specialized lexicons or general misspelling correction. Two important directions for improvement are to develop an improved corruption model and to extend the model to deal with multiple-word spelling errors.

## Acknowledgements

We acknowledge the support of Center for Machine Learning and Health at Carnegie Mellon University.

## Institutional Review Board (IRB)

This work does not require IRB approval.

## References

Eric Brill and Robert C Moore. An improved error model for noisy channel spelling correction. In *Proceedings of the 38th annual meeting of the associ-*

*ation for computational linguistics*, pages 286–293, 2000.

Fred J Damerau. A technique for computer detection and correction of spelling errors. *Communications of the ACM*, 7(3):171–176, 1964.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv.org*, October 2018.

DWYL. List of english words. https://github.com/dwyl/english-words, 2020. Commit on Oct 15, 2020.

Pieter Fivez, Simon Šuster, and Walter Daelemans. Unsupervised context-sensitive spelling correction of clinical free-text with word and character n-gram embeddings. In *BioNLP 2017*, pages 143–148, August 2017. doi: 10.18653/v1/W17-2317. URL https://www.aclweb.org/anthology/W17-2317.

Sai Muralidhar Jayanthi, Danish Pruthi, and Graham Neubig. Neuspell: A neural spelling correction toolkit. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 158–164, 2020.

Alistair E. W. Johnson, Tom J. Pollard, Lu Shen, Liwei H. Lehman, 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. doi: https://doi.org/10.1038/sdata.2016.35.

Mark D Kemighan, Kenneth Church, and William A Gale. A spelling correction program based on a noisy channel model. In *COLING 1990 Volume 2: Papers presented to the 13th International Conference on Computational Linguistics*, 1990.

Kenneth H Lai, Maxim Topaz, Foster R Goss, and Li Zhou. Automated misspelling detection and correction in clinical free-text records. *Journal of biomedical informatics*, 55:188–195, 2015.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, July 2020. doi: 10.18653/v1/2020.acl-main.703. URL https://www.aclweb.org/anthology/2020.acl-main.703.

Hao Li, Yang Wang, Xinyu Liu, Zhichao Sheng, and Si Wei. Spelling error correction using a nested rnn model and pseudo training data. *arXiv preprint arXiv:1811.00238*, 2018.

Xiangci Li, Hairong Liu, and Liang Huang. Context-aware stand-alone neural spelling correction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 407–414, 2020.

Chris J Lu, Alan R Aronson, Sonya E Shooshan, and Dina Demner-Fushman. Spell checker for consumer language (cspell). *Journal of the American Medical Informatics Association*, 26(3):211–218, 2019.

GB Moody, RG Mark, and AL Goldberger. Physionet: A research resource for studies of complex physiologic and biomedical signals. In *Computers in Cardiology 2000. Vol. 27 (Cat. 00CH37163)*, pages 179–182. IEEE, 2000.

Yifan Peng, Shankai Yan, and Zhiyong Lu. Transfer learning in biomedical natural language processing: An evaluation of bert and elmo on ten benchmarking datasets. In *Proceedings of the 2019 Workshop on Biomedical Natural Language Processing (BioNLP 2019)*, pages 58–65, 2019.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. 2018.

Patrick Ruch, Robert Baud, and Antoine Geissbühler. Using lexical disambiguation and named-entity recognition to improve spelling correction in the electronic patient record. *Artificial intelligence in medicine*, 29(1-2):169–184, 2003.

Herman D Tolentino, Michael D Matters, Wikke Walop, Barbara Law, Wesley Tong, Fang Liu, Paul Fontelo, Katrin Kohl, and Daniel C Payne. A umls-based spell checker for natural language processing in vaccine safety. *BMC medical informatics and decision making*, 7(1):1–13, 2007.

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.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, October 2020. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/2020.emnlp-demos.6.

## Appendix A. Data Processing

This section describes the data pre-processing procedure of the two real datasets in Section 5 and the reproduction of the spelling correction methods suggested in the dataset papers.

**MIMIC-III Misspelling Dataset** The MIMIC-III misspelling dataset is the only misspelling dataset based on the MIMIC-III clinical notes. However, there are several issues regarding their pre-processing code[1]. The code is based on an older version, v1.3, of the MIMIC-III database and splits processes text by the lines in the `NOTEEVENTS.csv` file, which leads to the risk to include the context from other notes in an example. After reviewed by a medical professional, we found that the correction labels of 30 examples are incorrect. We update these labels, some of which are multiple, and if an output is one of them, we marked it as correct. We will release our pre-processing code compatible with the latest version of the MIMIC-III database and revise the incorrect labels.

The baseline result in Table 2 is reproduced after running the word2vec training and hyper-parameter tuning. The numbers are not matched with Fivez et al. (2017) because of the revised labels and the randomness of the training and tuning.

**CSpell Spelling Error Dataset** The CSpell dataset contains various spelling errors: `Grammatical`, `Misspelling`, `Punctuation`, `RealWord`, `ToMerge`, and `ToSplit`. These spelling errors or their corrections can be multiple-word or contain non-alphabet characters or non-text entries, such as HTML entries. We only choose the examples in which the input word and the correction are single-word and alphabet-only.

As mentioned in the paper, CSpell software performs both detection and correction of spelling errors, and the input to the CSpell does not require the location of misspelled words. To find out which input word is detected, we use the output from the debug mode activated by the "`-d`" command-line flag. To make a fair comparison to our model, we excluded the CSpell test examples that the misspelled words are not detected by any of the detection modules of CSpell. Note that such filtering is not done on the CSpell training set to prevent our model from fitting to the CSpell's selection bias. As result, 409 and 574 examples are chosen from the training set and the test set of 1050 and 1924 examples, respectively.

| Hyper-param | Values |
|---|---|
| Early stopping | $\{25000n : n = 1, ..., 20\}$ steps |
| $C$ | $\{2.0, 5.0, 10.0, 20.0\}$ |
| $n$ | $\{0, 1, 2, 3, \infty\}$ |

Table 7: List of decoding hyper-parameters evaulated

The input texts of the CSpell dataset, consumer health questions, can contain multiple spelling errors, each of which constitutes an example. When we evaluate each misspelling, other misspellings in the same input text are corrected to clean the context and remove the interference of them. The output of CSpell software is evaluated only based on whether the typo words have been corrected properly, regardless of the other words.

## Appendix B. Training, Tuning and Evaluation

This section describes the training procedure, the hyper-parameter search, and the final evaluation of our spelling correction method. Our character-level language model is based on the BART (Lewis et al., 2020) implementation of Hugging Face's Transformer (Wolf et al., 2020). Both the Base and Large models are trained for 500k iterations on the MIMIC-III clinical notes with batch size 256. We trained the model on 4×NVIDIA A100 GPU 40GB. We follow the optimizer and the learning rate schedule same as the original BERT, except that the learning rate for the encoder part is reduced to one-tenth of the decoder since the encoder is finetuned from BlueBERT.

We tuned the hyper-parameters on the correction accuracy on the CSpell training set. Hyper-parameters of our method are only from the beam search decoding: the loss weight $C$ and the number of characters ahead $n$ in the corruption model. Also, for early stopping, we evaluated our method for every 25k steps of the training of the character-level language model. For the full model and each of the ablation studies, we did the grid search on all possible values of $C$, $n$, and training steps, summarized in Table 7. The selected hyper-parameter values are shown in the first eight columns of Table 8.

The time complexity of the beam search is proportional to the beam width $B$. $B = 30$ is chosen during the hyper-parameter tuning for faster search, and we

[1] https://github.com/clips/clinspell

| **Hyper-param** | CSpell Training | | | | | | | | Synthetic | |
|---|---|---|---|---|---|---|---|---|---|---|
| | LM only | | LM + Dict | | LM + ED | | LM+ED+Dict | | | |
| | Base | Large | Base | Large | Base | Large | Base | Large | Base | Large |
| Early stopping | 475k | 450k | 475k | 450k | 475k | 475k | 475k | 300k | 500k | 500k |
| $C$ | - | - | - | - | 5.0 | 5.0 | 5.0 | 5.0 | 5.0 | 5.0 |
| $n$ | - | - | - | - | 1 | 1 | 1 | 1 | 2 | 2 |

Table 8: The result of hyper-parameters search on different datasets and settings

use $B \in \{30, 300\}$ when evaluating our method on the CSpell test set and the Clinspell set for better correction outputs.

## Appendix C. Subgroup Analysis by UMLS Semantic Type

This section describes the grouping procedure of subgroup analysis by UMLS Semantic Types. We first choose three subtrees from the hierarchy of the UMLS Semantic Types [2] with head nodes `substance`, `Pathologic Function`, and `Finding`. Each subtree becomes a subgroup of Semantic Types that represents a word category: "substance", "disease", and "symptom". The Semantic Types of each subgroup are as follows:

- Substance: `Substance`, `Pharmacologic Substance`, `Antibiotic`, `Biomedical or Dental Material`, `Biologically Active Substance`, `Hormone`, `Enzyme`, `Vitamin`, `Immunologic Factor`, `Receptor`, `Hazardous or Poisonous Substance`, `Organic Chemical`, `Amino Acid, Peptide, or Protein`, `Inorganic Chemical`, `Element, Ion, or Isotope`, `Body Substance`, `Food`

- Disease: `Pathologic Function`, `Disease or Syndrome`, `Mental or Behavioral Dysfunction`, `Neoplastic Process`

- Symptom: `Finding`, `Laboratory or Test Result`, `Sign or Symptom`

Also, we made "other" subgroup for examples that do not belong any of three subgroups.

Then, we grouped examples into Subgroups with their correction words. Given an example, we retrieved CUIs (Concept Unique Identifiers) of its correction word and related Semantic Types by quering to UMLS API [3]. If any of the example's Semantic Types fall into a subgroup's Semantic Types, then we put the example into the subgroup. Note that examples can belong to more than one subgroup since a word can have multiple UMLS Semantic Types. For example, a word "depression" fall into both "disease" and "symptom" subgroups.

## Appendix D. Results in Unsupervised Setting

This section describes the procedure to generate the MIMIC-III synthetic misspelling dataset and the hyper-parameter tuning results on it.

Due to the limited amount of public dataset of clinical typo, we build a dataset of synthetically generated misspellings. From the MIMIC-III clinical notes, we randomly choose words that are in our reference dictionary and corrupt them with random operations. To corrupt words, up to two operations of character addition, deletion, substitution, or transposition can be applied. As a result, we generated 10k examples of misspellings. In the 10k examples of syntactically generated data, 10% are unchanged from the original word, 45% are modified with a single operation (insertion, deletion, or substitution), and 45% are modified with two operations. As a result, out of 8970 error examples, 3199 and 5771 are real and non-word errors, respectively.

We performed the hyper-parameter search on this synthetic dataset, as we did with the CSpell training set. Since it takes much computation to evaluate CIM on the synthetic dataset, we did not search for early stopping and choose the final language model. The grid search was performed on the other hyper-parameters, $C$ and $n$. The last two columns of Table 8 show the results.

[2] https://www.nlm.nih.gov/research/umls/META3_current_semantic_types.html

[3] We use the latest version of UMLS, which is 2021AA

## Appendix E. Spelling Correction Example

Figure 4(*a*) to 5(*d*) show the beam search results of our spelling correction model on several examples of the CSpell test set. For each example, we display the top 10 beam candidates (out of 300) of our Base and Large model. Our model combines the language model score (`LM`) and the corruption model score (`ED`) and normalizes it to get the final beam score (`Score`).

For easy cases such as Figure 4(*a*) and 5(*a*), both the language model and the corruption model gives a high score to the correct output. For the typo of a complex terminology like Figure 4(*b*), there might be an incorrect word ("proctectomy") that has a smaller edit distance than the correct word ("prostatectomy"), but the language model gives a high score to the correct word.

Also, we report some failure cases of our method. Figure 4(*c*) and 5(*c*) shows a failure case where the language model fails. In Figure 4(*c*), although both "having" and "facing" fit the context, the language model grants a much lower score to "facing", which makes "having" be the top candidate by the Base model. In Figure 4(*d*), although the correct word "tightening" gets the higher LM score, the corruption model gives a lower score it. This implies that the corruption model needs to be improved to give a higher score to the correct word in such cases.

```
 [Input]
Typo / Correct : prpostectomy / prostatectomy
Context: "my husband had a simple retropubic prpostectomy 6 weeks ago. he can not walk wit. My husband had a simple r
etropubic prostatectomy. he can not walk without the walker, he can not move his right leg alone"
 [Output]
Base/475k/B300/ED1(1)        Score      LM      ED | Large/300k/B300/ED1(1)      Score      LM      ED
prostatectomy              -1.1696 -0.0982 -1.0714 | prostatectomy             -1.1057 -0.0342 -1.0714
periostectomy              -1.5619 -0.8476 -0.7143 | proctectomy               -1.4805 -0.6472 -0.8333
proctectomy                -1.8191 -0.9858 -0.8333 | appendectomy              -2.2858 -0.3628 -1.9231
periosteotomy              -2.0782 -1.0068 -1.0714 | proctostomy               -2.2902 -0.6235 -1.6667
parotidectomy              -2.1398 -0.3541 -1.7857 | postectomy                -2.2912 -0.9275 -1.3636
appendectomy               -2.2063 -0.2832 -1.9231 | proctocolectomy           -2.3218 -0.4468 -1.8750
postectomy                 -2.2858 -0.9221 -1.3636 | cystoprostatectomy        -2.3307 -0.2254 -2.1053
preostectomy               -2.4425 -2.0578 -0.3846 | prostatotomy              -2.3428 -0.8043 -1.5385
parathyroidectomy          -2.4522 -0.2299 -2.2222 | nephrectomy               -2.3527 -0.2694 -2.0833
pylorectomy                -2.4765 -0.8098 -1.6667 | papillectomy              -2.3949 -0.8565 -1.5385
```

($a$) A success example

```
 [Input]
Typo / Correct : syntoms / symptoms
Context: "bad huge blood clots during her menstrual cycles after she was prescribed Ocella for birth control. Also th
ese syntoms worsened after she gave birth. This has been happening for a year now. should she see discuss this"
 [Output]
Base/475k/B300/ED1(1)        Score      LM      ED | Large/300k/B300/ED1(1)      Score      LM      ED
symptoms                   -1.2953 -0.1842 -1.1111 | symptoms                  -1.2335 -0.1223 -1.1111
systems                    -2.5961 -1.3461 -1.2500 | syndromes                 -2.4693 -0.9693 -1.5000
syndromes                  -2.6394 -1.1394 -1.5000 | systems                   -2.4899 -1.2399 -1.2500
syndrome                   -2.8220 -1.1553 -1.6667 | syndrome                  -2.6580 -0.9913 -1.6667
symptom                    -2.8439 -0.9689 -1.8750 | sensations                -2.6691 -0.3964 -2.2727
sensations                 -2.9676 -0.6948 -2.2727 | bottoms                   -2.7527 -0.8777 -1.8750
symptom's                  -2.9678 -1.4678 -1.5000 | symptom's                 -2.7878 -1.2878 -1.5000
syncopes                   -3.0207 -1.3540 -1.6667 | symptom                   -2.8031 -0.9281 -1.8750
symptomatics               -3.1481 -0.8404 -2.3077 | symptomatics              -2.9592 -0.6516 -2.3077
syncope                    -3.1973 -1.3223 -1.8750 | sensors                   -3.0031 -1.1281 -1.8750
```

($b$) A success example

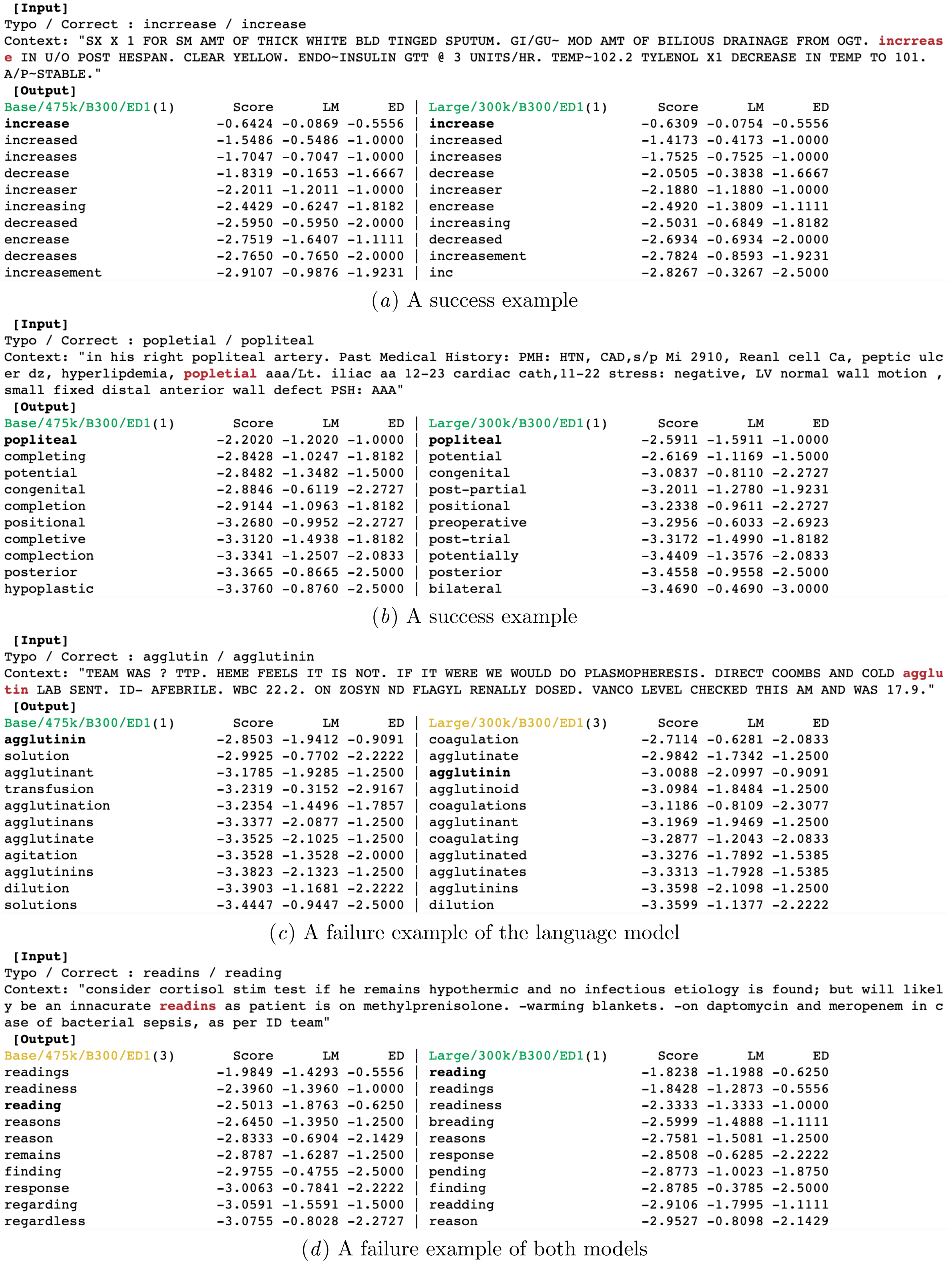
```
 [Input]
Typo / Correct : faceing / facing
Context: "A FEMALE AGED PATIENT OF 65 YEARS OLD IS suffering IN POLYUREA CONDITION SHE IS NOT A PATIENT OF diabetes S
HE faceing A PROBLEM OF EXCESSIVE URINE AT NIGHT TO GET UP NO change IN THE COLOR CHANG BLOOD IN URINE BURNING FEVER"
 [Output]
Base/475k/B300/ED1(2)        Score      LM      ED | Large/300k/B300/ED1(1)      Score      LM      ED
having                     -3.2793 -1.1364 -2.1429 | facing                    -2.8995 -2.1852 -0.7143
facing                     -3.3089 -2.5947 -0.7143 | having                    -3.0808 -0.9379 -2.1429
feeling                    -3.3493 -1.4743 -1.8750 | causing                   -3.2121 -1.3371 -1.8750
hacking                    -3.3700 -2.1200 -1.2500 | receiving                 -3.2722 -1.2722 -2.0000
causing                    -3.3907 -1.5157 -1.8750 | feeling                   -3.3320 -1.4570 -1.8750
happening                  -3.4183 -1.4183 -2.0000 | baseline                  -3.3624 -1.1402 -2.2222
falling                    -3.4725 -2.2225 -1.2500 | falling                   -3.3726 -2.1226 -1.2500
finding                    -3.5307 -1.6557 -1.8750 | raising                   -3.3795 -1.5045 -1.8750
passing                    -3.5499 -1.6749 -1.8750 | takeing                   -3.4534 -2.2034 -1.2500
baseline                   -3.5644 -1.3422 -2.2222 | happening                 -3.4825 -1.4825 -2.0000
```

($c$) A failure example of the language model

```
 [Input]
Typo / Correct : tighting / tightening
Context: "abdominal aorta aneurysm. I have an aneurysm (2.6cm for the last 5yrs last checked about a year ago experie
ncing pain, tighting in abdomen and chest should I be going to emergency room."
 [Output]
Base/475k/B300/ED1(3)        Score      LM      ED | Large/300k/B300/ED1(2)      Score      LM      ED
lighting                   -1.8270 -1.2715 -0.5556 | lighting                  -1.8186 -1.2631 -0.5556
righting                   -1.9277 -1.3721 -0.5556 | tightening                -1.8317 -0.9226 -0.9091
tightening                 -1.9883 -1.0792 -0.9091 | righting                  -1.9845 -1.4290 -0.5556
nighting                   -2.1890 -1.6335 -0.5556 | fighting                  -2.2743 -1.7188 -0.5556
tincting                   -2.3586 -1.2475 -1.1111 | nighting                  -2.2746 -1.7191 -0.5556
sighting                   -2.3814 -1.8258 -0.5556 | lightening                -2.4226 -1.0589 -1.3636
bighting                   -2.4258 -1.8702 -0.5556 | tightens                  -2.4664 -1.3553 -1.1111
tingling                   -2.4553 -0.7886 -1.6667 | sighting                  -2.5331 -1.9775 -0.5556
dighting                   -2.5457 -1.9901 -0.5556 | tightened                 -2.5589 -1.0589 -1.5000
tightness                  -2.5537 -0.5537 -2.0000 | rightening                -2.5678 -1.2042 -1.3636
```

($d$) A failure example of the corruption model

Figure 4: Beam search decoding results for several examples of the CSpell test set. For each example, we display the top 10 beam candidates. The column next to the candidate (`Score`) shows the final beam score for each candidate.



```
 [Input]
Typo / Correct : incrrease / increase
Context: "SX X 1 FOR SM AMT OF THICK WHITE BLD TINGED SPUTUM. GI/GU~ MOD AMT OF BILIOUS DRAINAGE FROM OGT. incrreas
e IN U/O POST HESPAN. CLEAR YELLOW. ENDO~INSULIN GTT @ 3 UNITS/HR. TEMP~102.2 TYLENOL X1 DECREASE IN TEMP TO 101.
A/P~STABLE."
 [Output]
Base/475k/B300/ED1(1)        Score      LM      ED | Large/300k/B300/ED1(1)      Score      LM      ED
increase                   -0.6424 -0.0869 -0.5556 | increase                  -0.6309 -0.0754 -0.5556
increased                  -1.5486 -0.5486 -1.0000 | increased                 -1.4173 -0.4173 -1.0000
increases                  -1.7047 -0.7047 -1.0000 | increases                 -1.7525 -0.7525 -1.0000
decrease                   -1.8319 -0.1653 -1.6667 | decrease                  -2.0505 -0.3838 -1.6667
increaser                  -2.2011 -1.2011 -1.0000 | increaser                 -2.1880 -1.1880 -1.0000
increasing                 -2.4429 -0.6247 -1.8182 | encrease                  -2.4920 -1.3809 -1.1111
decreased                  -2.5950 -0.5950 -2.0000 | increasing                -2.5031 -0.6849 -1.8182
encrease                   -2.7519 -1.6407 -1.1111 | decreased                 -2.6934 -0.6934 -2.0000
decreases                  -2.7650 -0.7650 -2.0000 | increasement              -2.7824 -0.8593 -1.9231
increasement               -2.9107 -0.9876 -1.9231 | inc                       -2.8267 -0.3267 -2.5000
```

(*a*) A success example

```
 [Input]
Typo / Correct : popletial / popliteal
Context: "in his right popliteal artery. Past Medical History: PMH: HTN, CAD,s/p Mi 2910, Reanl cell Ca, peptic ulc
er dz, hyperlipdemia, popletial aaa/Lt. iliac aa 12-23 cardiac cath,11-22 stress: negative, LV normal wall motion ,
small fixed distal anterior wall defect PSH: AAA"
 [Output]
Base/475k/B300/ED1(1)        Score      LM      ED | Large/300k/B300/ED1(1)      Score      LM      ED
popliteal                  -2.2020 -1.2020 -1.0000 | popliteal                 -2.5911 -1.5911 -1.0000
completing                 -2.8428 -1.0247 -1.8182 | potential                 -2.6169 -1.1169 -1.5000
potential                  -2.8482 -1.3482 -1.5000 | congenital                -3.0837 -0.8110 -2.2727
congenital                 -2.8846 -0.6119 -2.2727 | post-partial              -3.2011 -1.2780 -1.9231
completion                 -2.9144 -1.0963 -1.8182 | positional                -3.2338 -0.9611 -2.2727
positional                 -3.2680 -0.9952 -2.2727 | preoperative              -3.2956 -0.6033 -2.6923
completive                 -3.3120 -1.4938 -1.8182 | post-trial                -3.3172 -1.4990 -1.8182
complection                -3.3341 -1.2507 -2.0833 | potentially               -3.4409 -1.3576 -2.0833
posterior                  -3.3665 -0.8665 -2.5000 | posterior                 -3.4558 -0.9558 -2.5000
hypoplastic                -3.3760 -0.8760 -2.5000 | bilateral                 -3.4690 -0.4690 -3.0000
```

(*b*) A success example

```
 [Input]
Typo / Correct : agglutin / agglutinin
Context: "TEAM WAS ? TTP. HEME FEELS IT IS NOT. IF IT WERE WE WOULD DO PLASMOPHERESIS. DIRECT COOMBS AND COLD agglu
tin LAB SENT. ID- AFEBRILE. WBC 22.2. ON ZOSYN ND FLAGYL RENALLY DOSED. VANCO LEVEL CHECKED THIS AM AND WAS 17.9."
 [Output]
Base/475k/B300/ED1(1)        Score      LM      ED | Large/300k/B300/ED1(3)      Score      LM      ED
agglutinin                 -2.8503 -1.9412 -0.9091 | coagulation               -2.7114 -0.6281 -2.0833
solution                   -2.9925 -0.7702 -2.2222 | agglutinate               -2.9842 -1.7342 -1.2500
agglutinant                -3.1785 -1.9285 -1.2500 | agglutinin                -3.0088 -2.0997 -0.9091
transfusion                -3.2319 -0.3152 -2.9167 | agglutinoid               -3.0984 -1.8484 -1.2500
agglutination              -3.2354 -1.4496 -1.7857 | coagulations              -3.1186 -0.8109 -2.3077
agglutinans                -3.3377 -2.0877 -1.2500 | agglutinant               -3.1969 -1.9469 -1.2500
agglutinate                -3.3525 -2.1025 -1.2500 | coagulating               -3.2877 -1.2043 -2.0833
agitation                  -3.3528 -1.3528 -2.0000 | agglutinated              -3.3276 -1.7892 -1.5385
agglutinins                -3.3823 -2.1323 -1.2500 | agglutinates              -3.3313 -1.7928 -1.5385
dilution                   -3.3903 -1.1681 -2.2222 | agglutinins               -3.3598 -2.1098 -1.2500
solutions                  -3.4447 -0.9447 -2.5000 | dilution                  -3.3599 -1.1377 -2.2222
```

(*c*) A failure example of the language model

```
 [Input]
Typo / Correct : readins / reading
Context: "consider cortisol stim test if he remains hypothermic and no infectious etiology is found; but will likel
y be an innacurate readins as patient is on methylprenisolone. -warming blankets. -on daptomycin and meropenem in c
ase of bacterial sepsis, as per ID team"
 [Output]
Base/475k/B300/ED1(3)        Score      LM      ED | Large/300k/B300/ED1(1)      Score      LM      ED
readings                   -1.9849 -1.4293 -0.5556 | reading                   -1.8238 -1.1988 -0.6250
readiness                  -2.3960 -1.3960 -1.0000 | readings                  -1.8428 -1.2873 -0.5556
reading                    -2.5013 -1.8763 -0.6250 | readiness                 -2.3333 -1.3333 -1.0000
reasons                    -2.6450 -1.3950 -1.2500 | breading                  -2.5999 -1.4888 -1.1111
reason                     -2.8333 -0.6904 -2.1429 | reasons                   -2.7581 -1.5081 -1.2500
remains                    -2.8787 -1.6287 -1.2500 | response                  -2.8508 -0.6285 -2.2222
finding                    -2.9755 -0.4755 -2.5000 | pending                   -2.8773 -1.0023 -1.8750
response                   -3.0063 -0.7841 -2.2222 | finding                   -2.8785 -0.3785 -2.5000
regarding                  -3.0591 -1.5591 -1.5000 | readding                  -2.9106 -1.7995 -1.1111
regardless                 -3.0755 -0.8028 -2.2727 | reason                    -2.9527 -0.8098 -2.1429
```

(*d*) A failure example of both models

Figure 5: Beam search decoding results for several examples of the MIMIC-III misspelling dataset. For each example, we display the top 10 beam candidates. The column next to the candidate (`Score`) shows the final beam score for each candidate.