Noburp on Reddit: Expanding the Lexicon for Retrograde Cricopharyngeal Dysfunction (RCPD)

Adi Mukundan

I. INTRODUCTION

Retrograde cricopharyngeal dysfunction (RCPD), also called "no-burp syndrome," is a rare and underdiagnosed disorder that is characterized by an inability to burp due to dysfunction of the cricopharyngeus muscle. Due to this dysfunction in the upper esophageal sphincter (UES), there is a failure to vent gas through the esophagus causing bloating, gurgling sounds, and excessive flatulence. This results in varieties of symptoms impacting quality of life, physical pain, psychological distress, and general well being.

The disease was first reported in 2019 in Dr. Bastian's paper [1]. Since then there has been an increasing amount of studies and clinical trials on the disease. However, much is still unknown about the disease and a large scale study such as the one in this paper has never been done before. Social media in the form of Reddit lends itself to this task due to the large amount of data regarding RCPD available on `r/noburp`.

A. The data and data collection

Due to changes to Reddit's policies in 2023, public access to the Pushshift Reddit API was shut down. For this study, we obtained monthly Reddit dump files from the community hosted and compiled Academic Torrents site. From the full data obtained, we kept content from `r/noburp`.

`r/noburp` is a community centered on Retrograde Cricopharyngeal Dysfunction (called noburp due to the frequency and relevance of this term as a symptom of the disease), the community features discussions on the condition, symptoms, medical treatments/diagnostics, self treatment strategies, comorbidities, and various other discussions about life with the condition.

Unique authors	7,865
Posts	16,197
Comments	107,672
Date range	March 2014 – February 2025

B. Background data and manual analyses

This work builds on existing knowledge of the lexicon used in clinical settings to describe the disease RCPD. As such, it is necessary to begin the study using the knowledge of clinicians experienced in the language used to describe and diagnose the condition. Two clinicians from the University of Iowa Department of Otolaryngology conducted manual analyses of posts and comments in the corpus to provide ground truth data on what is used to describe RCPD by users on the subreddit.

The two clinicians manually annotated 1,786 documents (where a document is a post or a comment) from 49 different users. The clinicians annotated all terms and phrases in the document they deemed relevant to the description or experience of RCPD. This process resulted in the identification of 2,074 unique terms (381 unigrams, 916 bigrams, and 777 trigrams). Where a unigram is a single token (e.g., *bloated*), a bigram is a two token phrase (e.g., *chest pressure*), and a trigram is a three token phrase (e.g., *unable to burp*). This annotated set serves as the seed terms and phrases used in the method of this paper. In the rest of the paper this set of initial seed terms will be referred to as S_0. Where $S_0 = \{s_0^0, ...s_0^{2,073}\}$

> *Example excerpt from an annotated post.:*
>
> I'm also not as gurgly or bloated these past few days. However I did feel kinda nauseated with low appetite today.

Only the highlighted terms were selected by clinicians as RCPD-related.

C. Prior work and contributions of this work

The DUI paper [2] introduces a word embedding method to discover semantically similar terms in a corpus of Reddit text. The DUI method builds on the ideas in the original word2vec paper [3], embeddings trained on Reddit capture the discourse in substance use and recovery communities. In DUI, the method was used to expand a seed vocabulary of known substance use and

recovery terminology to identify slang, morphological variants, and related concepts to expand what is known about how drug use and recovery is described online.

We adapt the DUI approach to `r/noburp`. Starting from the clinician curated seed lexicon, we train word2vec embeddings on the subreddit corpus and retrieve nearest neighbors for each seed. The resulting neighborhood captures domain slang, morphological variants, and symptom/treatment terminology specific to RCPD.

We modify the method to reduce noise by introducing a semantic filtering step that keeps only candidates with a valid relation to their seed. Using the structure of WordNet synsets [4], we retain candidates only if they satisfy one of the following: (i) *hyponym/hypernym (is-a)* - a specific term vs. its more general class (e.g., *cricopharyngeal muscle* is a hyponym of *muscle*); (ii) *holonym/meronym (whole-part)* - a whole vs. its part (e.g., *pharyngeal region* is a holonym of the *upper esophageal sphincter*); (iii) *cohyponyms (siblings under a shared hypernym)* - peer terms that share a parent category (e.g., *Botox* injection, *dilation*, and *myotomy* under *medical procedure*); (iv) *acronym/expansion (abbreviation)* - the candidate is an acronym which falls into the synset of the seed or vice versa (e.g., *EGD* for *esophagogastroduodenoscopy*); (v) *synonym* - terms that share the same meaning (e.g., *belch* and *burp* are synonyms). For each seed-candidate pair, a large language model (LLM) is prompted to judge which (if any) of these relations holds. The LLM returns a strict schema (accept: `["term"]`; reject: `[]`) to enable verification and guard against hallucinations.

II. Method

A. Preprocessing for word2vec training

Before training word2vec, it was necessary to pre process and tokenize the Reddit text. We accomplished this using a text processing pipeline merging off the shelf python packages and libraries for tokenization, spellchecking, lemmatization, and stoplisting from [5-7]. Manual modifications were made to ensure these worked on the RCPD and Reddit specific vocabulary on *r/noburp*. Bigram and trigram phrases were merged with underscores to preserve phrases during training using the word2phrase model from [3]. With an n gram generation threshold of 5 empirically chosen. This allowed the creation of clinically relevant n grams (e.g., chest pressure→chest_pressure, no burp→no_burp) while minimizing irrelevant n grams from being generated. These preprocessing steps reduce vocabulary fragmentation, preserve clinically meaningful phrases, and improve the quality of the word2vec training content.

B. Training word2vec model

We trained word2vec using the continuous bag-of-words (CBOW) architecture on the full, preprocessed `r/noburp` corpus of 123,869 documents. We used **300** embedding dimensions, a symmetric context window size of **7**, **5** training epochs, and a min count of **5**.

In CBOW model training, the word2vec model predicts a center word from its surrounding context. A window size of **7** means up to seven tokens on the left and seven on the right are used as context for each target word. The embedding dimensions of **300** sets the size of each word vector. Epochs are full passes over the training corpus. The min count parameter prunes very rare tokens (frequency < 5), reducing noise and vocabulary fragmentation so the remaining vectors are trained on more reliable statistics.

We selected CBOW empirically because, for the corpus and objectives in this study, it returned tighter neighborhoods (i.e., more closely related terms) around clinician seed terms compared to the skipgram architecture. The choices of **300** dimensions and **5** epochs and window size of **7** follow prior practice in the DUI study [2] and empirical observations, while the min count of **5** reflects the setting recommended in the original word2vec work [3].

C. Embedding and expanding seed terms

We expand each seed by retrieving nearest neighbors in the embedding space $\mathbb{V}^{300}$ of the model using *cosine similarity*. For vectors $\mathbf{u}, \mathbf{v} \in \mathbb{V}^{300}$,

$$\cos(\theta) = \frac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{u}\|\|\mathbf{v}\|}$$

The word2vec embedding maps each term to a vector in a space where the vector's orientation encodes meaning. Similarity between two terms is then computed as the cosine of the angle between their vectors, so semantics can be represented by the geometry where closer angles mean a higher semantic similarity. Cosine is preferred over raw Euclidean distance because in high dimensions, Euclidean distances concentrate (nearest and farthest neighbors become similarly far) and also reflect vector norm differences often tied to term frequency, whereas cosine is length invariant and focuses on direction. We therefore rank by cosine and keep only the 25 nearest terms with another threshold

of 0.4 cosine similarity. Both of these set empirically through testing to keep uninformative terms in the expansion to a minimum. The following table shows an example expansion of the term *control* at three different cosine similarity thresholds, *control* is used in the context of `r/noburp` to refer to the practice of controlling and suppressing burps. The table shows the relevance dropping below the 0.4 threshold as well as the removal of relevant terms from the expansion above the 0.4 threshold, these observations informed the decision to choose the 0.4 threshold.

Cosine similarity threshold	Candidates
0.4	command [0.503], controllable [0.503], burp [0.501], suppress [0.478], natural [0.463], automatic [0.428], involuntary [0.419], relax [0.408], gradually [0.406], conscious [0.403], moon [0.401]
0.0	command [0.503], controllable [0.503], burp [0.501], suppress [0.478], natural [0.463], automatic [0.428], involuntary [0.419], relax [0.408], gradually [0.406], conscious [0.403], moon [0.401], satisfy [0.398], consistent [0.397], obsess [0.385], force [0.384], proper [0.379], bend [0.375], skill [0.375], belch [0.370], uncontrollable [0.369], still [0.368], manage [0.365], replicate [0.363], uncontrolled [0.362], learn [0.361]
0.8	

TABLE I: Expansion of the seed term "control" at three cosine similarity thresholds with cosine similarities of each candidate to the seed in brackets

The following algorithms show the expansion of a single term as well as the expansion of the entire seed set.

Algorithm 1 Expand one seed with word2vec + cosine

Require: seed s_i; word2vec model M (vocab V); $k = 25$, $\tau = 0.40$
Ensure: term list $T_i = \mathcal{E}_{k,\tau}(s_i)$ (up to k terms with cosine $> \tau$)

if $s_i \notin V$ **then**
  return [ ]
end if
$C \leftarrow \text{MOST\_SIMILAR}(M, s_i, \text{topn} = k)$
$T_i \leftarrow \{\, t \mid (t, c) \in C,\ c \geq \tau \,\}$
return T_i

Algorithm 2 One expansion round over a seed set $\rightarrow$ JSON map

Require: seed set S_i; model M; $k = 25$, $\tau = 0.40$
Ensure: JSON object J with entries $s \mapsto [t_1, \ldots, t_m]$

$J \leftarrow \{\}$
for each $s \in S_i$ **do**
  $T \leftarrow \mathcal{E}_{k,\tau}(s)$ ▷ Alg. 1: list of terms
  $J[s] \leftarrow T$ ▷ JSON entry: seed $\rightarrow$ array of terms
  $S_{i+1} \leftarrow S_{i+1} \cup \{\, t \mid t \in T \,\}$ ▷ grow the next seed set
end for
return J

After this filtered expansion the algorithm returns a Json object with a mapping from initial seed term to list of candidates. Due to noise and imperfect embeddings there remain terms unrelated to the seed terms in the candidates of that seed term. This table shows a handpicked example of this.

Seed	Candidate [cos]	Why unrelated
`control`	`moon`	No lexical relation, likely cooccurrence due to limited size of training data.

TABLE II: Illustrating false positives from raw cosine neighbors (before semantic filtering).

III. Semantic filtering

To address unrelated neighbors that survive cosine ranking, we add a semantic filter inspired by WordNet synsets [4]. We retain a candidate for a seed only if it stands in one of the following relations to the seed: (i) *hyponym/hypernym (is-a)*-a specific term vs. its more general class (e.g., *cricopharyngeal muscle* is a hyponym of *muscle*); (ii) *holonym/meronym (whole-*

part)-a whole vs. its part (e.g., *pharyngeal region* as a holonym of *upper esophageal sphincter*); (iii) *cohyponyms (shared hypernym)*-peer terms under a common parent (e.g., *Botox* injection, *dilation*, *myotomy* under *medical procedure*); (iv) *acronym/expansion (abbreviation)*-the candidate is an acronym of the seed or vice versa (e.g., *EGD* for *esophagogastroduodenoscopy*); and (v) *synonyms (same sense)*-terms that denote the same concept in context (e.g., *belch* and *burp*).

However, WordNet is a general English lexicon and does not cover many domain acronyms, multiword clinical terms, and colloquialisms common in `r/noburp`. This yields low recall if we rely on it directly. Two misses are shown below.

`EGD` ↔ `Esophagogastroduodenoscopy` (acronym of an upper respiratory examination not found in WordNet); `anesthetic` ↔ `propofol` (specific anasthetic used in Botox treatments for RCPD not in WordNet).

To recover such valid pairs on the type of vocabulary and domain specific language returned in the expansion on our corpus, we implement the structure and matching of WordNet with an LLM: for each seed-candidate, the model is prompted to judge whether any of the relations in the WordNet synset holds; only accepted pairs are kept. This method increases recall by leveraging the greater understanding of acronyms, medical specific language, and colloquialisms in social media LLMs have when compared to the WordNet database. The prompting framework is shown in figures 2 and 3 at the bottom of the document.

The LLM semantic filtering can be represented by these two equations which show the process for a single seed:candidate mapping and also for the entire set of expansion mappings generated in Algorithm 2. The output of algorithm 5 is the entire set of mappings after filtering as a set of new seed terms.

Algorithm 3 LLM semantic filter for a single seed - candidate

Require: JSON map $J : s \mapsto \mathcal{E}_{k,\tau}(s)$ from expansion; seed s

Ensure: filtered list $F(s)$ of accepted candidates for s

$S \leftarrow \text{keys}(J)$ ▷ all seeds (context)
$F(s) \leftarrow [\,]$
for each $t \in J[s]$ **do**
    $r \leftarrow \text{F}(s,\ t,\ S)$ ▷ LLM black box: sees seed, candidate, and all seeds
    if $r =$ accept **then**
        append t **to** $F(s)$
    end if
end for
return $F(s)$

Algorithm 4 LLM semantic filter for a seed - candidate list

Require: JSON map $J : s \mapsto T(s)$ from expansion; seed s

Ensure: filtered list $F(s)$ of accepted candidates for s

$S \leftarrow \text{keys}(J)$ ▷ all seeds (context)
$T \leftarrow J[s]$ ▷ entire candidate list for s
$F(s) \leftarrow \text{F}(s,\ T,\ S)$ ▷ LLM black box returns accepted terms
return $F(s)$

Algorithm 5 Apply LLM filter to all seeds

Require: JSON map $J : s \mapsto T(s)$ from expansion
Ensure: filtered JSON $J \mapsto F(s)$; next seed set S_{i+1}

$S \leftarrow \text{keys}(J)$
$\leftarrow \{\}$; $S_{i+1} \leftarrow S$
for each $s : [t] \in S$ **do**
  $F_s \leftarrow F(J,\ s)$ ▷ Alg. 3 or 4
  $J[s] \leftarrow F_s$
  $S_{i+1} \leftarrow S_{i+1} \cup \{\, t \mid t \in F_s \,\}$
end for
return $J,\ S_{i+1}$

These algorithms show the two modes in which the LLM semantic filter operates and how they are applied to the entire set of seeds and candidates to generate a new set of seed terms, called S_{i+1}. A common issue with LLMs are the prevalence of hallucinations in the output, this problem is limited by enforcing a strict schema in the generation of output decisions by the LLM, however errors are still possible. The table below shows the error rates of algorithm 3 and 4.

Algorithm	Queries to LLM	Errors
`Algorithm 3`	18,889	1 mismatch
`Algorithm 4`	2,074	79 mismatches

TABLE III: Algorithm, number of separate queries to the LLM, and number of errors

The mismatch errors are caught during the verification step after the LLM query is processed. The strict output schema is enforced and errors are shown in detail in table 4 at the bottom of the document. Other errors, such as the LLM deciding that two unrelated terms are semantically related or two semantically related terms are not related through a hallucination error or lack of training data, are more difficult to catch and require detailed analysis of the outputs.

IV. Iterating seed term set

Using algorithm 5, it is possible to iterate the seed term set and uncover new semantically related seed terms in the corpus. The size of the seed term set is shown in the following figure. The results when running algorithm 5 using the algorithm 4 filtering is shown in the following .

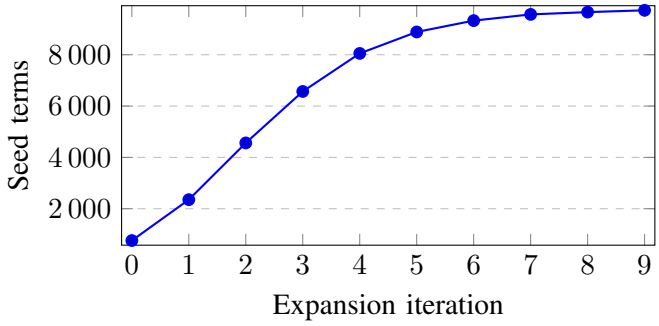

Fig. 1: Seed-term set size by expansion iteration using Algorithm 4 for expansion filtering.

The number of new seed terms appears to level off around 6–7 iterations. However, inspecting an individual chains reveals semantic drift. For example a selected seed, selected candidate flow, `exercise` → `workout` → `gym` → `marathon` → `half_marathon`. It can be seen how each concurrent term would fall into the WordNet synset of the previous term, however in our corpus the initial seed `exercise` is used in the *clinical* sense referring to shaker exercises and related maneuvers to facilitate belching/UES function in RCPD, not general fitness. This path seems to drift away from the RCPD domain. To curb drift, we can: (i) tighten the LLM filter by explicitly constraining the sense of ambiguous seeds (e.g., "*exercise* = therapeutic/clinical exercises; reject athletic/fitness uses"), (ii) retain only candidates that co-occur with RCPD anchor terms (e.g., `burp`, `gurgle`, `UES`, `Botox`) above a threshold in `r/noburp`, and (iii) raise the cosine threshold after early iterations. However, the fact that these terms are semantically close to the term *exercise* in the word2vec model, which has seen only the `r/noburp` corpus, indicates that these terms might contribute similar ideas of techniques/strategies for inducing burps and self-treating RCPD.

Further verification of the expanded seed sets will be carried out to flesh out these ideas.

Acknowledgements

- Overleaf.com, for the template

References

[1] R. W. Bastian and M. L. Smithson, "Inability to belch and associated symptoms due to retrograde cricopharyngeus dysfunction: Diagnosis and treatment," *OTO Open*, vol. 3, no. 1, p. 2473974X1983455, Jan 2019.

[2] Z. Prince, D. Jha, and R. Singh, "Dui: the drug use insights web server," *Bioinformatics*, vol. 37, no. 24, pp. 4895–4897, 06 2021. [Online]. Available: https://doi.org/10.1093/bioinformatics/btab461

[3] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient estimation of word representations in vector space,” *CoRR*, vol. abs/1301.3781, 2013. [Online]. Available: http://dblp.uni-trier.de/db/journals/corr/corr1301.htmlabs-1301-3781

[4] G. A. Miller, “WordNet: A lexical database for English,” in *Human Language Technology: Proceedings of a Workshop held at Plainsboro, New Jersey, March 8-11, 1994*, 1994. [Online]. Available: https://aclanthology.org/H94-1111/

[5] S. Bird, E. Klein, and E. Loper, *Natural language processing with Python: analyzing text with the natural language toolkit*. ”O’Reilly Media, Inc.”, 2009.

[6] B. Barr, “pyspellchecker: Pure python spell checking,” PyPI, The Python Package Index, 2024, version: 0.8.3. [Online]. Available: https://pypi.org/project/pyspellchecker/

[7] P. Qi, Y. Zhang, Y. Zhang, J. Bolton, C. D. Ma, X. Li, B. Zhang, and C. D. Manning, “Stanza: A python library for Natural Language Processing of many human languages,” *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, 2020. [Online]. Available: https://aclanthology.org/2020.acl-demos.26/

Case	Expected (strict)	Returned (caught)
Wrong key (Alg. 3; input: seed `antiemetics`, cand. `antihistamines`)	`"antiemetics"` → `"antihistamines"`	`"antihistamines"` → `"antihistamines"`
Duplicate item (Alg. 4; seed `70_unit`, list `["75u","50_unit",...]`)	`70_unit` → `[75u, 50_unit, ...]`	`70_unit` → `[75u, 50_unit, 75u, ...]`
Misspelled key (Alg. 4; seed `convince`)	`"convince"` → `[...]`	`"covice"` → `[...]`

TABLE IV: Schema-violation examples automatically rejected by the verifier (post-query strict schema check).

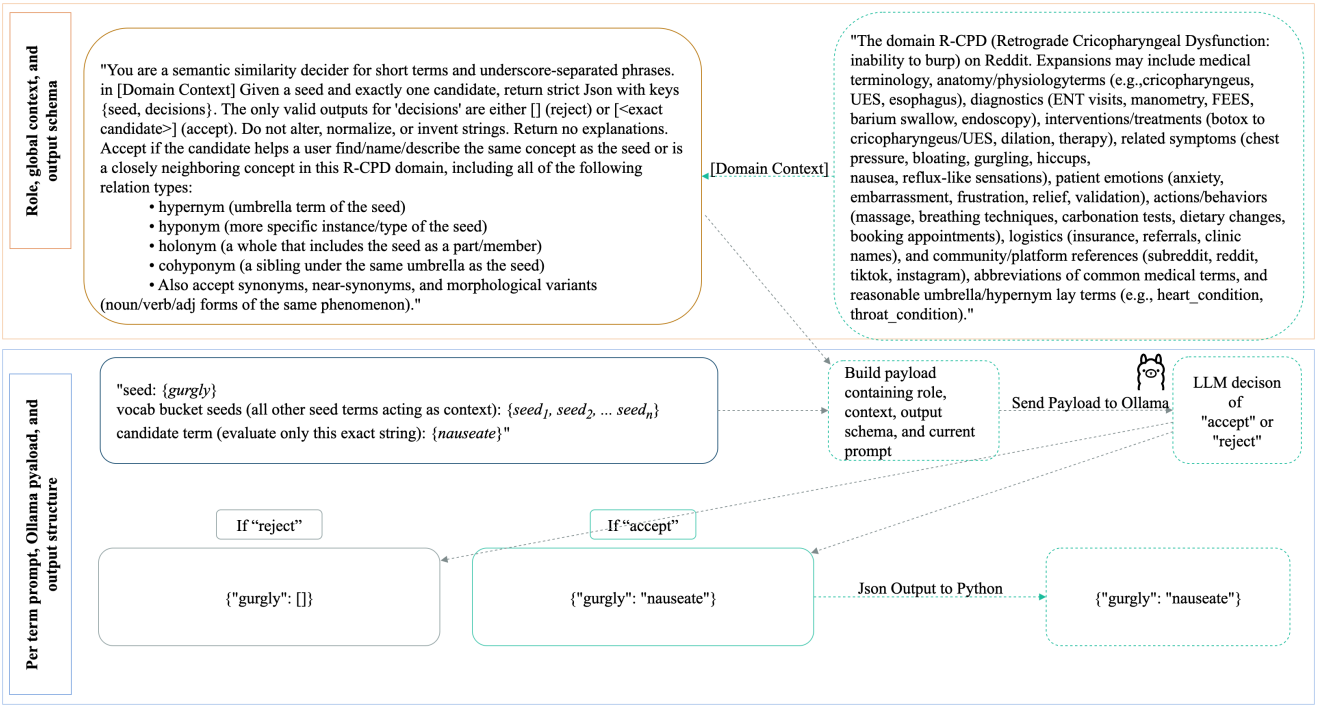

Fig. 2: LLM prompting framework for input of a single seed-candidate pair.

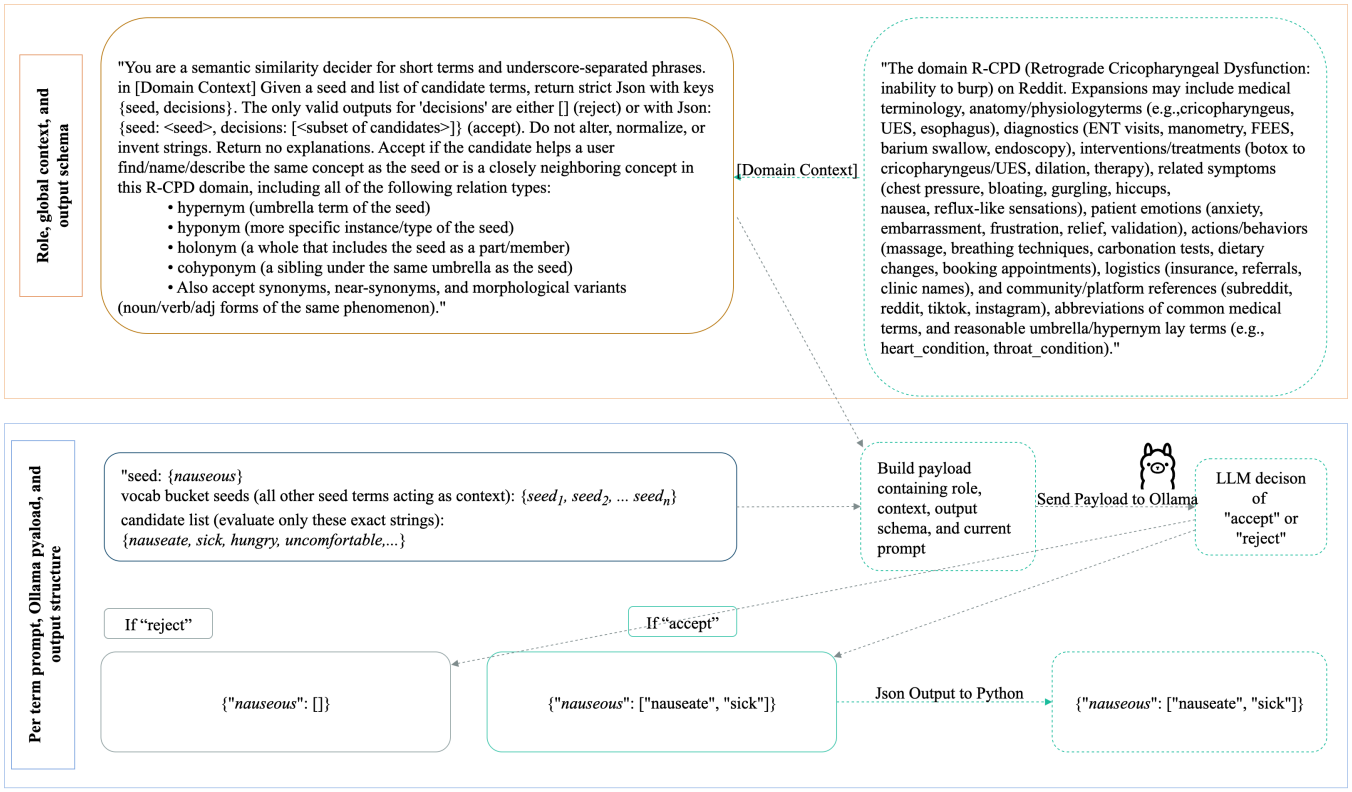

Fig. 3: LLM prompting framework applied to the full candidate list for a seed.