REVISITING MEDICAL CONCEPT NORMALIZATION: A COMPARATIVE ANALYSIS OF TRANSFORMER-BASED MODELS AND SEARCH ENGINE APPROACHES

by

Tugay Bilgis

A Thesis Submitted to The W.A. Franke Honors College

In Partial Fulfillment of the bachelor's degree
With Honors in

Computer Science

THE UNIVERSITY OF ARIZONA

2024

Approved by:

Dr. Steven Bethard

School of Information

TABLE OF CONTENTS

Abstract

This study examines the efficacy of medical concept normalization (MCN) by comparing transformer-based models and advanced search-based engines within the context of the 2019-n2c2-MCN dataset. Our approach focuses on replicating existing models, notably SAPBert, and evaluating their performance against custom-developed search engines optimized for MCN tasks. Through rigorous experimentation, we demonstrate how variations in indexing techniques and embedding strategies can impact the accuracy and robustness of MCN systems. The findings highlight critical insights into the scalability of current approaches and explore areas for potential improvement in efficiency and accuracy through future research.

1 Introduction

Advancements in clinical informatics have led to an accumulation of extensive textual data in medical records. However, these texts often contain diverse terminology and conceptual references due to varying linguistic expressions among healthcare providers. This variation poses significant challenges for automated systems that rely on a consistent understanding of medical terminology to function effectively. Medical concept normalization (MCN) addresses this challenge by linking textual mentions to their corresponding canonical forms within a structured ontology, thereby facilitating a unified representation of diverse expressions.

The task of MCN is crucial for numerous applications in medical informatics, including information retrieval, decision support systems, and clinical research, where the accuracy of entity recognition and normalization directly impacts outcome quality. Despite its importance, the process is complicated by the intrinsic complexity of medical language, characterized by abbreviations, synonyms, and highly technical terms. Additionally, the dynamic nature of medical lexicons, with continuously evolving terminology and the

introduction of new concepts, adds another layer of complexity to the task.

In this thesis, we aim to enhance medical concept normalization by critically evaluating the replication of existing state-of-the-art MCN methods and exploring more efficient approaches such as search engines. Specifically, we scrutinize the performance of transformer-based models, known for their efficacy across diverse linguistic environments, and compare them with advanced search-based engines that we create for this task. Through this comprehensive approach, we seek to advance the understanding of how these powerful models and search technologies can be fine-tuned and adapted to meet the unique challenges of MCN, ultimately enhancing their utility and reliability in clinical informatics applications.

2 Background

2.1 Unified Medical Language System (UMLS)

The Unified Medical Language System (UMLS) (Lindberg, Humphreys, and McCray 1993) is a crucial resource developed by the National Library of Medicine (NLM) to facilitate the integration of biomedical and health information across various computer systems. By consolidating numerous health and biomedical vocabularies and standards into a single framework, UMLS aids in the effective communication of medical concepts across different applications and datasets. Each concept in the UMLS is assigned an alphanumeric code called a Concept Unique Identifier (CUI). This identifier ensures that different terminologies referring to the same concept can be linked and recognized across various medical databases and applications, facilitating accurate and consistent information exchange.

UMLS is regularly updated and released in new versions to accommodate the latest medical knowledge and changes in medical terminology. Each version of UMLS may include updates to the concepts, terms, and relationships within its components, reflecting

advancements in medical science, modifications in terminology, or corrections from previous releases. Users must be aware of the version used in their applications as differences between versions can impact data consistency and interoperability. We are using multiple versions in this work and report the versions we use.

In our work, we leverage three essential vocabularies from the UMLS to ensure accurate and consistent medical terminology. Vocabularies in medical informatics are standardized sets of terms and definitions used to ensure consistent communication, documentation, and analysis of health-related information across different systems and databases. The vocabularies we use in this work include SNOMED CT (Systematized Nomenclature of Medicine—Clinical Terms), which is extensively used for clinical documentation; RxNorm, which provides standardized names for clinical drugs, facilitating interoperability across pharmacy management and drug interaction systems; and MSH (Medical Subject Headings), primarily used for indexing journal articles in the life sciences.

2.2 Related Work

SAPBert (Self-Alignment Pretraining for BERT) is a BERT model specifically trained using the Unified Medical Language System (UMLS) ontology to generate embeddings for medical concepts. This approach, introduced by Liu et al. (2021), focuses on enhancing the representation of biomedical entities through self-alignment pretraining techniques, where the model aligns medical terms with their semantic meanings as represented in UMLS. SAPBert is particularly notable for its ability to produce high-quality embeddings that facilitate improved performance in tasks such as medical concept normalization.

One of the most recent and comprehensive work on medical concept normalization comes from Xu and Miller (2022). In their work, they develop a neural-based model utilizing SapBert and a softmax layer. They comprehensively study different models and report the results of each model. Their novel approach achieves a better performance than the older models while having a simpler inference and more efficient approach.

The 2019 n2c2/UMass Lowell shared task on clinical concept normalization (MCN) (Luo, Henry, et al. 2020) saw participation from 33 teams deploying a variety of methods for medical concept normalization with the n2c2 dataset mentioned in Section 2.3. These teams utilized four primary approaches: cascading dictionary matching, cosine distance measures, deep learning techniques, and retrieve-and-rank systems.

The cascading dictionary matching approach was the most popular, employed by six of the top ten teams. This method typically started with exact matches followed by different matching strategies like edit distance or machine learning-based rankers for handling inexact matches. The winning team used a deep learning model that computed cosine similarity between SciBERT-generated embeddings of clinical text mentions and learnable vectors for each concept unique identifier (CUI) in the UMLS. Their model optimized these embeddings to align closely with the correct CUIs, effectively learning the contextual relationships within clinical narratives.

Overall, the task highlighted both the progress and challenges in the field of MCN, with the best-performing team achieving an accuracy of 85.26%, and the common difficulties centered around disambiguating acronyms, abbreviations, and complex multiword terms. This initiative not only pushed forward the state-of-the-art in MCN but also set a benchmark for future research in the domain.

The work by Zhang and Bethard (2023) focuses on geocoding, where location mentions in texts are matched with their geographical location. In their work, they propose a solution using a search engine such as Lucene Search, which provides an efficient method with better results. The objective of geocoding is similar to medical concept normalization, thus, we explore the same method of using a search engine for MCN.

2.3 Dataset

We used the 2019-n2c2-MCN dataset provided by the 2019 n2c2/OHNLP shared task track 3. This dataset uses the MCN corpus (Luo, Sun, and Rumshisky 2019), and it has a

	train	test
# of patient notes	50	50
# of medical terms	6684	6925
# of CUI-less terms	151	257
# of ambiguous terms	190	192
# of unique concepts	2331	2579

Table 1: Statistics for 2019-n2c2-MCN dataset

variety of medical terms (problems, treatments, tests, etc.) from the patient notes that the doctors have written. All the terms are mapped to one of 434,056 possible concepts in the SNOMED-CT and the RxNORM subset of UMLS version 2017AB.

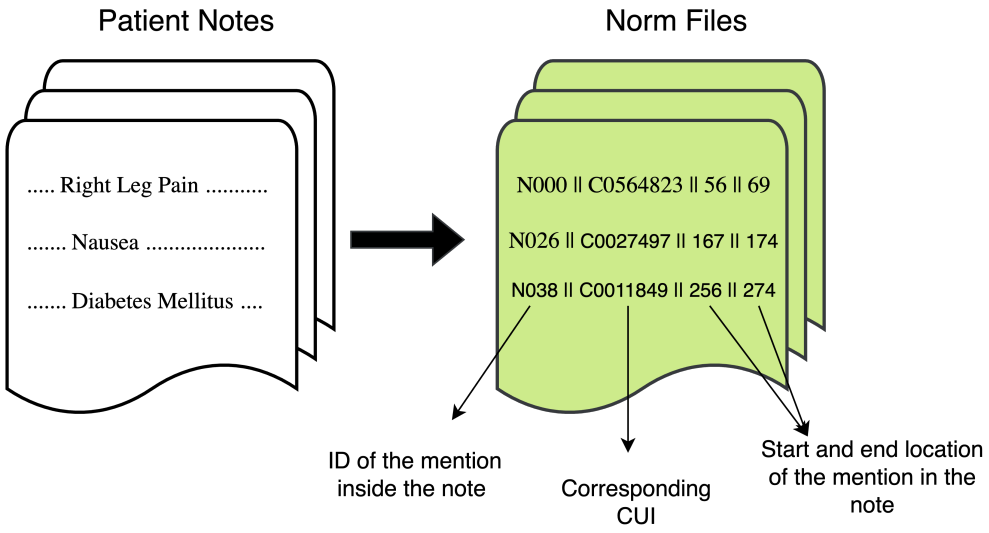

Figure 1: The visual description of the n2c2 dataset.

The format of the dataset is shown in Figure 1. All patient notes are in a .txt file and have a .norm file associated with it. Each line in the .norm files corresponds to one mention in the patient note. The starting and ending positions of the mentions are provided in these lines, as well as the corresponding CUI for the mention (UNK in the test set.)

Table 1 shows the statistics for this dataset. The numbers are derived from Xu and Miller (2022). There are 100 patient notes in total, 50 in train and 50 in test set. 2.8% of the terms are ambiguous, where a term is annotated with multiple concepts and the context is important to correctly map to a CUI, and 2.7% of the terms don't have a

corresponding CUI in the UMLS subset version 2017AB.

3 Methodology

Given a set of medical terms $\{t_1, t_2, \ldots, t_n\}$ and a set of CUIs from an ontology $\{c_1, c_2, \ldots, c_n\}$, the goal of medical concept normalization is to learn a function $f(t_i) = c_j$ that maps each term to their corresponding CUI.

Our experiments aimed to replicate the work by Xu and Miller (2022) and improve it with more efficient search and approximation algorithms. We additionally study the following settings: exact vs. fuzzy information retrieval, different neural network pooling methods, different hyperparameters for approximate nearest neighbor search, and different methods to create embeddings.

3.1 Replication study

The work by Xu and Miller (2022) evaluates various models on the 2019-n2c2-MCN dataset. We focus on replicating the results that were achieved with SAPBert (Liu et al. 2021) off-the-shelf. The purpose of this replication study is to analyze if the same results are achievable, and if they are, to develop a new model that would perform better.

SAPBert (Self-Alignment Pretraining BERT) is a BERT-based model that was fine-tuned on UMLS 2020AA. Given a medical term, the model generates an embedding with 768 dimensions.

To replicate Xu and Miller (2022)'s work, we use the SNOMED-CT and RxNORM subset of UMLS 2017AB. We create an embedding for each entry in this subset using SAPBert and store them. Then, we create embeddings for the terms in the train and test set of 2019-n2c2-MCN dataset and perform an approximate nearest neighbor search on the UMLS embeddings that were initially created. The CUI-less terms in the dataset are disregarded and are not considered while evaluating the model as SAPBert doesn't map

any concept as CUI-less (Xu and Miller 2022). Figure 2 visually explains this workflow.

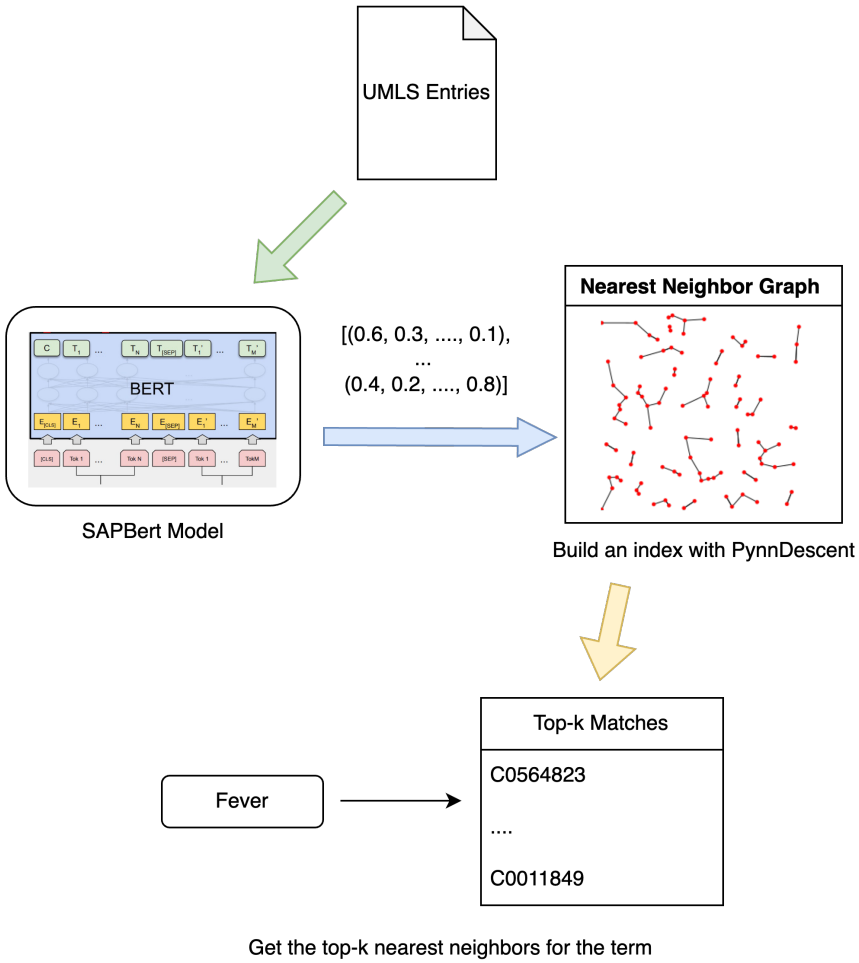

Figure 2: Visual workflow of the replication study

Due to the high-dimension and high number of embeddings, we employed an approximate nearest neighbor search for faster and more efficient performance. For the nearest neighbor approximation, we use the PyNNDescent library, which is implemented in Python and provides a fast approximate nearest neighbor search. It provides various customization options for the approximate search.

We build an index with the UMLS embeddings. Xu and Miller (2022) doesn't mention how exactly they perform their nearest neighbor search. The official PyNNDescent documentation [1] states that there is a trade-off between accuracy and efficiency when

[1] PyNNDescent Documentation

changing the index parameters. The main parameters of concern were: `n_neighbors`, `diversify_prob`, and `pruning_degree_multiplier`. The `n_neighbors` parameter specifies the number of neighboring data points each point in the dataset should consider during the initial construction of the nearest neighbor graph. It is set to 30 by default. The `diversity_prob` parameter is used to manage the diversity of the results in nearest neighbor searches. The default value of this parameter is 1.0. The `pruning_degree_multiplier` parameter controls the density of the nearest neighbor graph by scaling the number of connections each point maintains. It is set to 1.5 by default.

Except when otherwise noted, we set the parameters of PyNNDescent as: `n_neighbors = 100`, `diversify_prob = 0.0`, and `pruning_degree_multiplier = 3.0`. See Section 4.3 for experiments with other parameter settings.

3.2 SAPBert Robustness Studies

In addition to the replication study, we study how robust SAPBert is if we make changes in the data. Instead of using the UMLS 2017AB to generate embeddings and build the index, we experiment with using the UMLS 2023AA and adding the MSH subset, which reflects the common case when a user wants to use the most recent UMLS version rather than the older 2017 version used in Xu and Miller (2022). We use the default parameters of PyNNDescent when building the index in robustness experiments.

Xu and Miller (2022)'s work mentions that using average pooling yielded better results than using the CLS token. Thus, we also experiment with both approaches in our robustness study.

We also experiment with whether it is better to generate and index one embedding per CUI, or one embedding per medical term (resulting in multiple embeddings for the many CUIs that can be referred to by more than one term).

3.3 Building a Search Engine

Using a search engine such as Lucene search to index all the entities in a corpus and then training a neural network model to rerank the top-k results returned from the search engine is a powerful way to map geological names mentions to their correct locations on the map (Zhang and Bethard 2023). This method is efficient as there is no need to train or use an additional neural network model to generate word embeddings for the geological mentions. Following the same idea, we experiment with Whoosh, a search library in Python.

Whoosh is a fast search engine library that is implemented in pure Python. It uses the Okapi BM25 ranking function by default. It is an efficient library and does not require any compilers, it will run anywhere Python runs.

Whoosh offers a variety of analyzers, which is a function that returns a generator of tokens. The tokens are the units that get indexed and different analyzers will create a different set of tokens for a given string, which can make the search more accurate but also slower.

Whoosh also supports Fuzzy Search. Fuzzy search matches any similar term within a certain Damerau-Levenshtein edit distance and can help match the terms that have been misspelled or the terms that do not appear the same in UMLS.

We index the SNOMED-CT and RxNORM subsets of UMLS using Whoosh to create our search engine. We start our experimentation by only using exact matching, as that is most efficient, and we build several other indices with different analyzers later on. We experiment with the following analyzers: Standard analyzer, N-Gram analyzer, and Stemming Analyzer. We incorporate fuzzy search into the search engine in our experiments.

The workflow for the search is given in 3. The initial experiments were conducted with the assumption that the gold CUI is already known. If the predicted CUI from one analyzer is wrong, we switch to the other one as shown in the workflow. This allows us to analyze the maximum potential of the search engine mechanism and if it can potentially

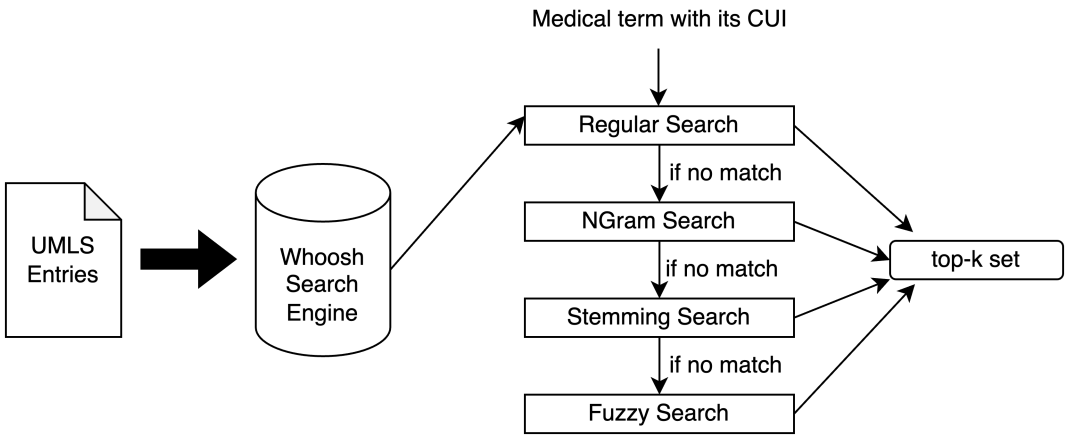

Figure 3: Visual workflow for the Whoosh Search Engine

outperform SAPBert. We experiment with both UMLS subsets as mentioned in sections 3.1 and 3.2.

4 Results and Discussion

We evaluate the models with the recall metric over the size of the nearest neighbor/match set that the models return: Recall@1, Recall@20, and Recall@100. Both the approximate nearest neighbor and the Whoosh index can be modified to return the top-n matches. In our case, we pick n as 1, 20, and 100. For each medical term, we check if these sets contain the corresponding CUI. If it is in the set, we consider it as a match. Then, the formula for Recall@N is given as

$$Recall@N = \frac{\text{\# of matches in the n-length set}}{\text{\# of medical terms}}$$

We evaluate the models on the train set for most of our experiments. The training set is not used for any training in our experiments.

Method	R@1	R@20	R@100
Whoosh Index w/ Exact Match	0.634	0.761	0.787
+ N-gram and Stemming Analyzer, Fuzzy Match			
SAPBert (Avg. Pooling)	0.703	0.861	0.913
SAPBert (Avg. Pooling) on Test Set	0.687	0.856	0.913

Table 2: Results of different searching techniques on the n2c2 dataset with the SNOMEDCT-US and RxNORM subset of UMLS 2017AB. R@k = Recall @ k.

4.1 Replication Results

Table 2 shows the results of the replication study and the results with the Whoosh index on the same UMLS subset.

The SAPBert (off-the-shelf) results, which are in rows 2 & 3, we achieve differ from the results that Xu and Miller (2022) report in their paper. In their experiments, they use the percentage of entity mentions that were correctly normalized as the evaluation metric, which corresponds to Recall@1 in our evaluation. They report a Recall@1 score of 0.8397 for dev set and 0.8277 for test set with SAPBert off-the-shelf, while we achieve 0.703 on train set.

Specific to this replication study, we additionally evaluate the SAPBert model on the test set to make sure our results do not differ because of a different set. We achieve a lower score, 0.688, when using the test set instead of the train set.

As described in Section 3.1, we follow the same approaches and the only component that potentially differs is the nearest neighbor search. We use a nearest-neighbor approximation, while Xu and Miller (2022) might be using an exact neighbor search. To investigate if this is the issue, we randomly pick a patient note and perform an exact nearest neighbor search with the cdist function from scipy [2]. The selected patient note has 119 medical terms.

We compare the closest neighbor prediction between the approximate and exact nearest neighbor search. Out of the 119 terms, only 1 term had a mismatch between the approximate and exact nearest neighbor prediction. We conclude that using an

[2]Scipy cdist documentation

Method	R@1	R@20	R@100
Whoosh Index w/ Exact Match	-	0.60	-
+ N-gram and Stemming Analyzer, Fuzzy Match	-	0.74	0.78
SapBert (CLS Token)	-	**0.833**	**0.883**
SapBert (Avg. Pooling)	0.62	0.831	0.88
SapBert (Avg. Pooling) w/ Single embedding for one CUI	0.058	0.258	0.308

Table 3: Results of different searching techniques on the n2c2 dataset with the SNOMEDCT-US, MSH, and RxNORM of UMLS 2023AA. Missing fields indicate that the value was not calculated for that method. R@k = Recall @ k.

approximate nearest neighbor search is not the reason for such a difference in Recall@1 scores between our experiments and the work of Xu and Miller (2022).

As we struggled to replicate the results for SAPBert(off-the-shelf), we were limited in making progress that would contribute to the existing solution and introduce a novel approach to medical concept normalization.

4.2 Results with Whoosh

The first row of Table 2 shows the results for the search engine as described in Section 3.3 using Whoosh.

The Whoosh index performs significantly worse than the SAPBert. Though it is indeed faster and more efficient to use Whoosh, however, the results are worse and not on par with SAPBert. Unlike the work by Zhang and Bethard (2023) on geological entities, using a search engine for medical entities does not provide a better approach than using a Transformer-based model.

4.3 Robustness Results

Table 3 provides results of the various robustness experiments described in Sections 3.2 and 3.3.

Using Whoosh with only exact matching yields a Recall@20 score of 0.60 with the UMLS 2023AA subset. This is very far from the Recall@20 score of the SAPBert model

in the replication study. When the N-gram and Stemming Analyzers are added alongside the Regular Analyzer, and additionally the Fuzzy match, the Recall scores are close to the results of the Whoosh index created with UMLS 2017AB without the MSH subset.

The SapBert recall scores for the robustness experiments are lower than the replication study results. This is likely due to having more entries in the UMLS with the addition of MSH subset, and since we are using the default parameters for the PyNNDescent index. While the Recall@20 and Recall@100 are closer to the replication study results, there is a higher gap between the Recall@1 scores. This indicates that predicting the right CUI as the nearest neighbor is more sensitive to different parameters than predicting the top-20 or top-100 nearest neighbors. Interestingly, we get higher recall scores with CLS token whereas Xu and Miller (2022) got higher scores when using average pooling.

The last entry in Table 3 shows the results of using a single embedding for one CUI, where we concatenate all the terms related to CUI to create an embedding. This method yielded the worst results out of all the experiments. It can be interpreted that SapBert does not do so well in embedding generation when multiple terms are combined.

5 Conclusion

This thesis has explored the challenging domain of medical concept normalization by examining the efficiency of different models and techniques on the 2019-n2c2-MCN dataset. Through the rigorous replication study and experiments with Whoosh and SAPBert, we have gained insights into the strengths and limitations of current approaches.

Our replication of Xu and Miller (2022) using SAPBert provided a foundation to test the robustness and scalability of this model with newer UMLS versions and alternative indexing techniques. Despite efforts, replicating the original results was challenging, underscoring the sensitivity of these models to subtle variations in experimental setups and parameter configurations. The additional robustness studies with SAPBert using

UMLS 2023AA and various indexing parameters revealed that while performance varies, it consistently fell short of our expectations when the model was extended beyond its initial configuration.

The experiments with Whoosh introduced a potential direction for future work, although they did not outperform the more sophisticated neural approaches. The use of different analyzers and fuzzy matching provided incremental improvements, suggesting that enhancements in search engine configurations could be a worthwhile exploration for specific applications.

Future research should prioritize enhancing the adaptability and accuracy of medical concept normalization techniques. This could involve the development of hybrid models that integrate the strengths of transformer-based and search engine-based methodologies, or the creation of more sophisticated indexing mechanisms that can more effectively manage the complexity of medical lexicons.

References

Lindberg, D. A., B. L. Humphreys, and A. T. McCray (1993). "The Unified Medical Language System". In: *Methods of Information in Medicine* 32.4, pp. 281–291. DOI: 10.1055/s-0038-1634945.

Liu, Fangyu et al. (June 2021). "Self-Alignment Pretraining for Biomedical Entity Representations". In: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Ed. by Kristina Toutanova et al. Online: Association for Computational Linguistics, pp. 4228–4238. DOI: 10.18653/v1/2021.naacl-main.334. URL: https://aclanthology.org/2021.naacl-main.334.

Luo, Yen-Fu, Sam Henry, et al. (Sept. 2020). "The 2019 n2c2/UMass Lowell shared task on clinical concept normalization". In: *Journal of the American Medical Informatics Association* 27.10, 1529–e1. ISSN: 1527-974X. DOI: 10.1093/jamia/ocaa106. eprint: https://academic.oup.com/jamia/article-pdf/27/10/1529/39739985/ocaa106.pdf. URL: https://doi.org/10.1093/jamia/ocaa106.

Luo, Yen-Fu, Weiyi Sun, and Anna Rumshisky (2019). "MCN: A comprehensive corpus for medical concept normalization". In: *Journal of Biomedical Informatics* 92, p. 103132. ISSN: 1532-0464. DOI: https://doi.org/10.1016/j.jbi.2019.103132. URL: https://www.sciencedirect.com/science/article/pii/S1532046419300504.

Xu, Dongfang and Timothy Miller (2022). "A simple neural vector space model for medical concept normalization using concept embeddings". In: *Journal of Biomedical Informatics* 130, p. 104080. ISSN: 1532-0464. DOI: https://doi.org/10.1016/j.jbi.2022.104080. URL: https://www.sciencedirect.com/science/article/pii/S153204642200096X.

Zhang, Zeyu and Steven Bethard (July 2023). "Improving Toponym Resolution with Better Candidate Generation, Transformer-based Reranking, and Two-Stage Resolution". In: *Proceedings of the 12th Joint Conference on Lexical and Computational Semantics (*SEM 2023)*. Ed. by Alexis Palmer and Jose Camacho-collados. Toronto, Canada: Association for Computational Linguistics, pp. 48–60. DOI: 10.18653/v1/2023.starsem-1.6. URL: https://aclanthology.org/2023.starsem-1.6.