Aman Anand, 2026, 14:5
ISSN (Online): 2348-4098
ISSN (Print): 2395-4752

International Journal of Science, Engineering and Technology
An Open Access Journal

Health Sathi AI

Aman Anand[1], Harsh Tiwari[2], Gaurav Kumar[3]
B.Tech Computer Science And Engineering,
Mahatma Gandhi Mission College Of Engineering & Technology, Noida, India

Abstract— Hospitals produce a huge amount of free-text material every day, from prescriptions to discharge notes, yet very little of it can be searched or analysed in a structured way. Medical concept recognition, which finds clinical mentions in text and ties them to standard identifiers, is the usual route to unlock it. This paper reviews three families of approaches (dictionary and rule-based systems, the MedCAT toolkit, and transformer-based models) and then examines one practical case in detail: adapting a pre-trained MedCAT model to prescription-style text containing Indian medicine names, carried out during an internship at C-DAC Noida. Instead of relying on retraining alone, the work concentrated on the model's concept database, adding Indian brand names, spelling variants and synonyms, and fine-tuning on 500 synthetic records. On a small in-house test set, synonym mapping raised symptom recognition from 40.00% to 86.36%, disease recognition from 61.54% to 86.54% and medicine recognition from 21.84% to 43.68%. A coverage check also showed that the baseline model matched only 229 of 1,808 Indian medicines (12.67%). The review concludes that the quality of the concept vocabulary deserves as much attention as the choice of model, and it lists the evaluation gaps that still need to be closed.

Keywords— medical NLP, named entity recognition, entity linking, MedCAT, UMLS, concept database, Indian medicines, synthetic clinical data

I. INTRODUCTION

Walk into any hospital outpatient department and you will find the same thing: notes written in a hurry, abbreviations that only the writer fully understands, and prescriptions in which a drug may appear under its brand name, its generic name, or some local spelling of either. All of this is free text. It is useful to the doctor who wrote it, but a computer cannot count it, filter it or learn from it until the diseases, symptoms and medicines inside it have been picked out and mapped to something standard.

That is the job of medical concept recognition. It has two halves that are easy to confuse. Named entity recognition (NER) decides which stretch of text is a medical mention. Entity linking then decides which concept that mention refers to, usually a Concept Unique Identifier (CUI) in the Unified Medical Language System (UMLS) [1]. Tools that do both are often called NER+L systems.

Most published work is built around English text from Western hospitals. Indian prescriptions are different in ways that matter to a model: brand names such as Pantop or Betnovate are far more common than generic names, spelling is inconsistent, and local terms appear next to clinical ones. A model that scores well on a public benchmark can still miss most of the medicines in a real Indian prescription. We ran into exactly this problem during an internship at the Speech and Natural Language Processing (SNLP) department of C-DAC Noida, where a pre-trained MedCAT model [2] left many medicines unrecognised.
This paper has two aims.

The first is a compact review of how the main families of tools approach concept recognition, written for readers who are new to the area. The second is a case study of what we actually did to repair the gaps in one of them, reported with its limits left visible. Our contributions are:

- a short comparison of rule-based, MedCAT-style and transformer-based recognition, with the trade-offs that mattered in practice;
- a four-way classification of why an entity goes missing from the output, which turned out to be the most useful debugging idea of the project;

© 2026 Aman Anand, This is an Open Access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly credited.

- measured results on vocabulary expansion for Indian medicines, including the baseline coverage figure; and
- a candid list of what the evaluation does not show.

II. BACKGROUND

1. From Raw Text To Standard Concepts

Consider the sentence "Patient has fever and cough, prescribed Pantop." A recogniser must first mark "fever", "cough" and "Pantop" as medical mentions. Linking then attaches each one to a concept. UMLS pulls many medical vocabularies into one resource and gives every concept a CUI [1].

Each concept also carries one or more semantic types, written as TUI codes. For instance, T047 stands for Disease or Syndrome, T184 for Sign or Symptom and T121 for Pharmacologic Substance. These codes are what allow a downstream application to sort entities into categories such as Drug, Disease and Symptom.

The distinction between detection and linking is more than vocabulary. When an entity is missing from the final output there are at least three possible reasons: the span was never detected, it was detected but could not be linked, or it was linked and then thrown away by later filtering. Each has a different cure, so diagnosing the right one comes before any fix.

2. How Medcat Is Organised

MedCAT is an open-source toolkit built on spaCy [2], [3]. Text is tokenised, candidate terms are looked up in a Concept Database (CDB), and when a term could mean several things the words around it are compared with a stored context vector for each candidate. The similarity score doubles as a confidence value. A model pack bundles the CDB, a vocabulary of word vectors, configuration settings and the spaCy model into one ZIP file that can run on an ordinary computer.

Two details of the CDB shaped our work. First, it maps surface names to CUIs (name2cuis) and CUIs back to their names, preferred name, semantic types and training counts. Second, and this surprised us at first, the model knows only what is inside that loaded CDB. A medicine that is listed in a spreadsheet sitting next to the model is invisible to it.

3. Measuring Quality

Entity recognition is normally judged with precision, recall and the F1-score. Precision asks how many reported entities are right; recall asks how many of the entities that should have been found were actually found; F1 combines the two. The percentages reported in our case study are closest to recall, because we counted expected entities that were found and did not separately count wrong ones. We return to this limitation in Section VII.

III. APPROACHES TO CONCEPT RECOGNITION

1. Dictionary And Rule-Based Systems

MetaMap [4] and cTAKES [5] were among the first widely used tools. They rely on linguistic analysis and dictionary look-up against UMLS. QuickUMLS [6] trades some of that linguistic machinery for fast approximate string matching. These systems are transparent and predictable, which clinicians appreciate, but they lean heavily on lexical matching. A term that is not in the dictionary, or is spelled differently, is simply missed, and ambiguous words are resolved with limited context.

2. Context-Aware Linking: Scispacy And Medcat

scispaCy [7] supplies spaCy models trained on biomedical text, with entity linking to UMLS. MedCAT [2] goes a step further by learning a context representation for each concept, which lets it separate two concepts that share a name. It can also be trained without labels on a large body of text and later fine-tuned on local data. Crucially for our setting, its CDB can be edited directly, so a missing concept is a data problem and not a research problem.

3. Transformer-Based Models

Pre-trained language models such as BERT [8], and domain versions like BioBERT [9] and ClinicalBERT [10], have pushed NER accuracy considerably on benchmark corpora. They usually need labelled data to fine-tune, and they tag spans without necessarily assigning a stable standard identifier. Where output must be repeatable, auditable and tied to UMLS codes, a linker with an editable vocabulary remains attractive.

4. Large Language Models As A Reference

General-purpose large language models (LLMs) also read prescriptions well, largely because they have seen many brand names. In our project we did not use one as the production extractor. We used it as a probe: whatever an LLM found that MedCAT missed became a targeted test case. The practical differences are summarised in Table I.

TABLE I. Practical comparison of the approaches

Aspect	Rule / dictionary	MedCAT	LLM
Standard IDs	Yes (UMLS)	Yes (CUIs)	Usually no
Same input, same output	Yes	Yes	Not always
Add a new term	Edit dictionary	Edit CDB	Prompt / retrain
Runs locally	Yes	Yes	Often no
Brand-name knowledge	Only if listed	Only if in CDB	Broad

I. IV. CASE STUDY: ADAPTING MEDCAT TO INDIAN PRESCRIPTION TEXT

A. *A. Diagnosing why entities were missing*

We began by running the pre-trained model on sample clinical and conversation files and saving the results as JSON and Excel. Entities were often absent from the Excel output, so before touching the model we printed the raw MedCAT output and compared it, step by step, with what the application finally exported. That comparison separated model problems from application problems and led to the four groups in Table II.

TABLE II. Why an entity goes missing

Group	What happens	Fix
Not in CDB	Name or synonym is absent	Add concept or synonym
Detected, not linked	Span found, context score too low	Train for better context
Linked, then filtered	Model returned it; later logic dropped it	Correct filter or mapping
Export problem	Present but never written out	Correct export code

The third group was a real eye-opener. The application originally classified entities using only three hard-coded semantic types, so many valid disease and drug concepts were quietly discarded.

Once classification moved to the backend and used wider sets of TUI codes, the output improved without any change to the model itself.

5. Data Used

Real prescriptions are hard to obtain because of privacy rules, so we worked with three kinds of text. We looked at several public sources, including MIMIC-IV, MTSamples and MedMentions [12], and settled on PubTator because it carries standard concept identifiers. First, 8,784 title-and-abstract documents were extracted from the PubTator corpus [11] and used for unsupervised training of MedCAT. Second, we wrote 500 synthetic records in the style of an Indian-English prescription, each combining complaints, examination findings, investigations, medicines, advice and follow-up. Third, we prepared 152 short context sentences (51 for symptoms, 51 for diseases and 50 for drugs) for category-wise testing. The 500 records and the PubTator documents were used for training, while the short sentences served mainly for testing.

6. Expanding The Cdb

The central intervention was vocabulary work. We studied how the CDB stores names and then added Indian brand names, generic names, common spelling variants and synonyms without disturbing existing concepts. Examples included Betnovate, Budecort, Benadryl Syrup, Bavencio Injection, Bleocip, Anovate, Avomine, Pantop and Ciplox. A small experiment confirmed the idea: the sentence "The patient was prescribed Pantop." produced nothing at first, but Pantop was extracted correctly as soon as its concept was registered in the loaded CDB. A final update added 20 new names and produced a refreshed model pack.

Prescriptions write the same product in many ways, so we adopted a few simple normalisation rules: compare names in lower case, store a brand with and without its dosage form (Benadryl and Benadryl Syrup), add a common misspelling as a synonym, and map brand and generic names to a single concept. Look-ups were done in levels, trying the exact term first and a modest variant (such as a trailing "s") second. This kept matching forgiving without opening the door to loose, false-positive matches.

7. Fine-Tuning And Structured Output

Starting from the pre-trained model pack rather than from scratch, we fine-tuned for several epochs on the synthetic records and the customised CDB. The motivation was practical: this keeps what the base model already knows and needs far less data. Finally, each extracted entity was assigned one of three categories, Drug, Disease or Symptom, using sets of semantic-type codes, and returned as JSON with its name, CUI and category.

A. *E. A fast and safe way to add a medicine*

Saving a full model pack takes a long time because the ZIP file is large. Doing it after every added medicine would leave the user waiting. We therefore split the action: the Excel catalogue and the in-memory CDB are updated immediately, so the medicine is recognised at once, while the slow save runs in a background thread guarded by a lock so that only one save happens at a time. Before the original pack is replaced, the saved ZIP is opened to confirm that it really contains the updated CDB.

8. Results

Table III shows how much of a reference list of 1,808 Indian medicines the baseline model recognised. Only 229 were matched, a direct coverage of about 12.67%. In other words, the largest source of error was not a subtle modelling weakness but plain absence from the vocabulary.

TABLE III. Baseline medicine coverage

Measure	Value
Medicines in reference list	1,808
Matched by baseline model	229
Missing	1,579
Direct coverage	about 12.67%

Table IV and Fig. 1 report recognition rates on our working test set before and after synonym mapping. Symptoms improved by 46.36 percentage points and diseases by 25.00 points, and medicine recognition doubled, though it still stays below 50%.

TABLE IV. Recognition before and after synonym mapping

Category	Before	After	Change (pp)
Symptoms	40.00%	86.36%	+46.36
Diseases	61.54%	86.54%	+25.00
Medicines	21.84%	43.68%	+21.84

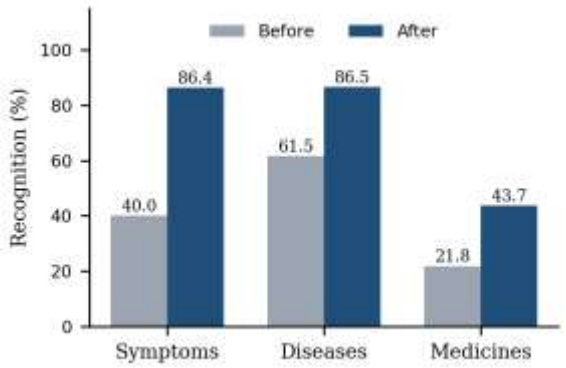


Fig. 1. Recognition before and after synonym mapping (project test set).

We want to be careful about how these numbers are read. They come from a small test set built during the project, they measure found-versus-expected entities rather than precision, and we did not isolate the separate contribution of PubTator training, fine-tuning and synonym mapping. They show that targeted vocabulary work helps a great deal on this data. They are not a general benchmark for MedCAT.

IV. DISCUSSION

1. Vocabulary Is Half The Model

The most consistent lesson was that a well-trained model still cannot find what its concept database does not contain. For the categories we targeted, adding the right names and synonyms moved the numbers more than we expected from training alone. Vocabulary engineering therefore deserves to be planned and budgeted as a core task and not treated as clean-up work.

2. A Spreadsheet Is Not A Cdb

A catalogue in Excel is a pleasant way to browse and review concepts, but only the CDB inside the loaded model changes what is extracted. The Pantop experiment made the point cleanly: the medicine was in

the catalogue and still went undetected until it was registered in the CDB.

3. Recall Must Be Balanced Against False Positives
Every added synonym widens the net. Words with both medical and everyday meanings are the risk. "Cold" is a disease in "common cold" and just a temperature in "cold weather", yet a careless synonym can make both appear in the output. We checked each new synonym against such paired sentences, and we would recommend that habit to anyone extending a CDB.

4. Negation Is Still Unsolved By The Base Pipeline
The base MedCAT pipeline recognises "diabetes" in "Patient has diabetes", "Patient has no diabetes" and "Diabetes was ruled out" alike. Telling presence from absence needs an additional meta-annotation component such as MetaCAT. For anything that feeds clinical decisions this is a serious limitation, and it is the first item we would tackle next.

5. Synthetic Data, Privacy And Responsible Use

Synthetic records let the project move forward without touching real patient information and let us deliberately include awkward medicines. Their weakness is limited variety, so a model meant for real use must eventually be tested on de-identified real documents checked by medical experts. More broadly, an entity recogniser is an assistive tool. Its output can be wrong for rare drugs, abbreviations and negated statements, and a qualified person should verify it before it influences care.

6. Engineering Lessons
Several delays had nothing to do with machine learning. A Windows application-control policy blocked a native spaCy file, which we traced through the Code Integrity logs and handed to the IT administrators because users could not change the policy. A version mismatch between spaCy and MedCAT removed an entity attribute we had been relying on. These episodes are a reminder to pin Python, spaCy and MedCAT versions and to keep backups of every model pack before editing its CDB.

7. Limitations And Future Directions
The evaluation in this paper has clear limits. The test set is small and was prepared by us, the figures approximate recall and say little about precision, the training data are partly synthetic, and in-memory changes still need a fully synchronised save to survive a restart. The agenda that follows from this is fairly direct:

1. build a manually verified gold-standard set of de-identified prescriptions and report precision, recall and F1 for each category;
2. keep growing the Indian medicine list, with brand-to-generic mappings and handling of abbreviations and spelling variation;
3. add negation and context handling, for example with MetaCAT;
4. version and back up the CDB automatically, and run regression tests after every vocabulary change so that new synonyms do not quietly lower accuracy.

V. CONCLUSION

Medical concept recognition is only as good as the vocabulary behind it. Our review of rule-based, MedCAT-style and transformer-based approaches suggests that no single family wins everywhere, and that a linker with an editable concept database is a sensible choice when identifiers, repeatability and local control matter. The case study shows what that editability buys: after systematic expansion of the CDB for Indian medicines and synonyms, recognition on our test set rose sharply for symptoms and diseases and doubled for medicines, while the 12.67% baseline coverage showed how far there is still to go. With a larger verified dataset and proper handling of negation, we expect this line of work to become a dependable base for processing Indian prescription text.

1. Acknowledgment
The authors thank C-DAC Noida and the SNLP department for the opportunity to work on this problem, and their colleagues and fellow interns for many helpful discussions.

REFERENCES

1. O. Bodenreider, "The Unified Medical Language System (UMLS): integrating biomedical terminology," Nucleic Acids Research, vol. 32, pp. D267–D270, 2004.

2. Z. Kraljevic, T. Searle, A. Shek, et al., "Multi-domain clinical natural language processing with MedCAT: the Medical Concept Annotation Toolkit," Artificial Intelligence in Medicine, vol. 117, p. 102083, 2021.
3. MedCAT documentation and source code. [Online]. Available: https://medcat.readthedocs.io/ and https://github.com/CogStack/MedCAT
4. A. R. Aronson, "Effective mapping of biomedical text to the UMLS Metathesaurus: the MetaMap program," in Proc. AMIA Symposium, 2001, pp. 17–21.
5. G. K. Savova, J. J. Masanz, P. V. Ogren, et al., "Mayo clinical Text Analysis and Knowledge Extraction System (cTAKES): architecture, component evaluation and applications," J. Am. Med. Inform. Assoc., vol. 17, no. 5, pp. 507–513, 2010.
6. L. Soldaini and N. Goharian, "QuickUMLS: a fast, unsupervised approach for medical concept extraction," in MedIR Workshop, SIGIR, 2016.
7. M. Neumann, D. King, I. Beltagy, and W. Ammar, "ScispaCy: fast and robust models for biomedical natural language processing," in Proc. 18th BioNLP Workshop, ACL, 2019.
8. J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," in Proc. NAACL-HLT, 2019.
9. J. Lee, W. Yoon, S. Kim, et al., "BioBERT: a pre-trained biomedical language representation model for biomedical text mining," Bioinformatics, vol. 36, no. 4, pp. 1234–1240, 2020.
10. E. Alsentzer, J. Murphy, W. Boag, et al., "Publicly available clinical BERT embeddings," in Proc. 2nd Clinical NLP Workshop, 2019.
11. C.-H. Wei, H.-Y. Kao, and Z. Lu, "PubTator: a web-based text mining tool for assisting biocuration," Nucleic Acids Research, vol. 41, pp. W518–W522, 2013.
12. S. Mohan and D. Li, "MedMentions: a large biomedical corpus annotated with UMLS concepts," in Automated Knowledge Base Construction (AKBC), 2019.