MedCAT v2: a modular, extensible architecture for clinical named entity recognition and linking under real-world privacy and compute constraints

Mart Ratas[1], Thomas Searle[1], Adam Sutton[1], Richard Dobson[1,2,3],
[1]The Department of Biostatistics & Health Informatics; Institute of Psychiatry, Psychology and Neuroscience; King's College London; London, United Kingdom,
[2]Institute of Health Informatics; University College London; London, United Kingdom,
[3]NIHR Mausley Biomedical Research Centre; London, United Kingdom,
Correspondence: mart.ratas@kcl.ac.uk

Abstract

MedCAT is an open-source framework for clinical named entity recognition and linking (NER+L) widely used in research and healthcare settings. We present MedCAT v2, a re-engineered version designed to improve modularity, extensibility, and maintainability while preserving the core functionality and performance of previous releases. The new architecture introduces a registry-based component system and a flexible pipeline that enables easy substitution of components, integration of alternative methods, and future expansion, including support for pre-trained components across the full NER+L and contextualisation workflow. This enables systematic exploration of clinical NER+L design trade-offs by evaluating different components in the pipeline. Evaluation across multiple public datasets shows equivalent or improved performance compared to earlier versions, with reduced integration overhead and improved runtime flexibility. The framework also supports optional extensions such as meta-annotation, relation extraction, providing a unified and reproducible environment for clinical NLP in real-world settings.

1 Introduction

Clinical free text (e.g., clinic letters, radiology reports, pathology notes, and discharge summaries) contains a substantial proportion of clinically relevant information that is not captured in structured electronic health records (EHR), limiting its secondary use for research, quality improvement, and decision support (Capurro et al., 2014; Kharrazi et al., 2018; Seinen et al., 2025). Advances in clinical natural language processing (NLP) now enable large-scale extraction of biomedical concepts from such data (Elvas et al., 2025). However, deploying such methods in routine clinical and research workflows introduces additional practical constraints: deployments often must run on-premises due to governance and privacy requirements (Van Bulck et al., 2024), may lack GPU-accelerated infrastructure (Wu et al., 2022), and require systems that can be maintained and adapted without substantial engineering overhead (Liu et al., 2025).

MedCAT (Medical Concept Annotation Toolkit) is an open-source framework for clinical named entity recognition and linking (NER+L) [1] that integrates large clinical ontologies such as SNOMED-CT and UMLS(Kraljevic et al., 2021), and also supports contextual meta-annotations (Mascio et al., 2020a; Agarwal et al., 2025b). The toolkit has been deployed across multiple institutions in the UK and internationally, enabling scalable extraction of structured clinical data in real-world settings (Mascio et al., 2020b; Noor et al., 2022; Au Yeung et al., 2024; Van Es et al., 2023).

Despite its success, operational use of MedCAT v1 highlighted several architectural limitations that restricted experimentation with alternative approaches. Exploring alternative approaches often required modifying or forking the codebase, increasing the maintenance burden. In addition, preserving backward compatibility while supporting diverse deployment environments introduced further complexity, particularly in resource-constrained clinical settings.

To address these challenges, we present MedCAT v2, a re-engineered modular architecture designed to support flexible configuration while preserving the proven core functionality of earlier releases. MedCAT v2 introduces a registry-based component system that enables interchangeable tokenizers, NER modules, and linkers, allowing alternative implementations to be integrated without modifying the core pipeline. This design supports systematic exploration of different NER+L configurations and facilitates experimentation with alternative modelling approaches.

[1] https://github.com/CogStack/cogstack-nlp/tree/main/medcat-v2

Proceedings of the 25th Workshop on Biomedical Language Processing (BioNLP 2026), pages 191–198

MedCAT v2 provides a sustainable foundation for clinical NLP by reducing integration overhead and exposing practical trade-offs between accuracy, throughput, and resource use. Our primary contributions are as follows:

- **Modular Architecture**: We introduce a registry-based component system for clinical NER+L that enables interchangeable tokenizers, NER modules, and linkers without modifying the core pipeline.
- **Performance-Throughput Framework**: We provide a redesigned infrastructure that allows for the systematic exploration of deployment trade-offs, supporting both lightweight CPU-only and high-throughput GPU-enabled environments.
- **Empirical Evaluation**: We demonstrate across multiple public datasets that this modular approach maintains or improves upon the performance of previous versions while offering significantly greater flexibility for real-world clinical deployment.

2 MedCAT v2 Architecture

2.1 Limitations of the MedCAT v1 architecture

Fig.1 demonstrates the monolithic architecture built around the spaCy framework. The components of the pipeline operated directly on *spaCy* objects such as documents, tokens, and spans, enabling a straightforward implementation and tight integration with the *spaCy* ecosystem.

However, this design introduced strong coupling between MedCAT's internal components and the underlying *spaCy* data structures. Changes to upstream processing steps, such as tokenization, required corresponding modifications throughout the rest of the pipeline to ensure compatibility with the expected document and span representations.

In addition, MedCAT v1 did not provide a mechanism for defining or registering alternative implementations of tokenizers or core components (e.g named entity recognition (NER) or entity linking (EL)). Exploring alternative approaches therefore required modifying or forking the codebase and integrating new implementations directly into the pipeline logic, increasing maintenance overhead and making experimentation with alternative methods difficult.

2.2 Modular and extensible pipeline architecture

To address these limitations, MedCAT v2 adopts a modular pipeline architecture designed around interchangeable components. Core stages of the pipeline (tokenization, named entity recognition (NER), and entity linking (EL)) are independently defined and registered, allowing alternative implementations to be integrated without modifying the rest of the system.

A lightweight *registry system* manages the available components and their implementations. This enables components to be swapped, extended, or replaced while maintaining a consistent pipeline interface. In addition to the core components, MedCAT v2 supports optional add-on modules that operate on the annotated document, such as MetaCAT for contextual meta-annotations and RelCAT for relation extraction (Agarwal et al., 2025a).

The registry architecture also enables support for external extensions: independent software packages that provide additional component implementations. These extensions can register their components with the MedCAT pipeline at runtime, allowing new functionality to be distributed separately from the core framework. For example, alternative linking approaches such as embedding-based linkers can be provided as external extensions while remaining fully compatible with the MedCAT pipeline.

To support reproducibility and deployment transparency, the framework tracks the provenance of each component used in a pipeline, including whether it originates from the core framework or from an external extension. This information is recorded in the model metadata, allowing pipelines to be fully described and reconstructed across different environments.

Fig. 2 describes the high level architecture of MedCAT 2.0. Components not explicitly discussed in this paper retain the same underlying implementation and behaviour as in MedCAT v1, and readers are referred to the original MedCAT publications for additional details on these components.

2.3 Example component implementations

The registry-based architecture allows different implementations of each pipeline stage to be configured depending on the requirements of a deployment. Components can be provided either by the core MedCAT library or by external extension pack-

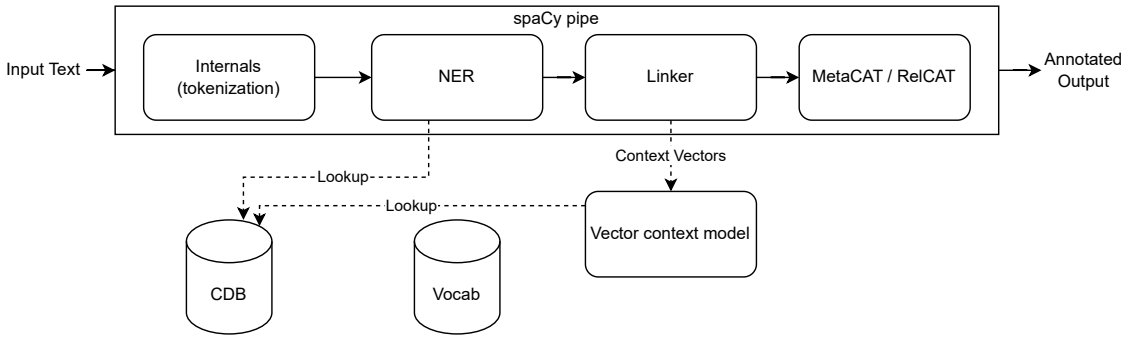

Figure 1: High-level architecture of MedCAT v1 showing the monolithic pipeline design built around *spaCy*.

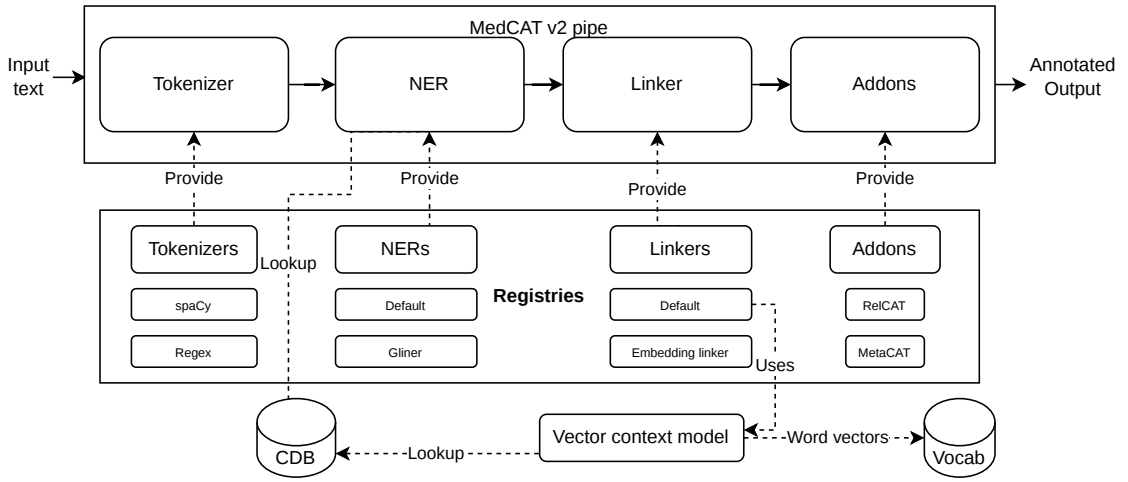

Figure 2: Modular registry-based architecture of MedCAT v2, illustrating interchangeable core components, optional add-on modules, and example registered implementations.

ages, enabling new functionality to be integrated without modifying the core system. For example, tokenization can be performed using either the default *spaCy*-based tokenizer or a lightweight regular-expression tokenizer designed for minimal environments. Similarly, alternative implementations of the NER and entity linking components can be used to balance trade-offs between computational cost, throughput, and predictive performance. To illustrate this flexibility, we highlight several component implementations available with MedCAT v2.

We demonstrate the improved flexibility through the following new components:

- **Regex-based tokenizer.** A lightweight tokenizer based on regular expressions designed for high-throughput or resource-constrained environments. While this approach reduces dependency overhead and improves processing speed, it may sacrifice some linguistic accuracy compared to more sophisticated tokenization methods
- **Faster linker.** A heuristic linker that resolves entities when a surface form maps unambiguously to a single concept, or when only one candidate concept treats the surface form as its primary name. By bypassing full disambiguation, this component substantially improves inference speed at the cost of reduced recall in ambiguous contexts.
- **Embedding-based linker.** A disambiguation component that leverages transformer based architectures (e.g. *sentence-transformers/all-MiniLM-L6-v2*) to improve concept selection for short or ambiguous mentions. In the configuration evaluated in this work, the linker uses static embeddings and provides modest improvements depending on the dataset and configuration. However, the architecture also supports training task-specific embedding representations for entity disambiguation, which

can substantially improve performance. This implementation is distributed as an independent MedCAT extension package on PyPI, demonstrating how external components can provide alternative implementations for core pipeline stages.

- *Gliner based NER*. A named entity recognition component based on GLiNER (Zaratiana et al., 2024), which can be used as an alternative to the default MedCAT vocabulary-driven NER pipeline. This component demonstrates how transformer-based external NER systems can be integrated into MedCAT v2 through the modular pipeline architecture. While the default MedCAT approach relies on vocabulary matching combined with normalization and spell-correction mechanisms, the GLiNER-based implementation enables experimentation with fully neural NER approaches within the same downstream linking pipeline. This is an NER step based on gliner (Zaratiana et al., 2024).

2.4 Backward compatibility and migration

To support a smooth transition for existing users, MedCAT v2 includes a legacy compatibility layer that allows models trained with MedCAT v1 to be loaded and used within the new framework with no user intervention. This ensures that previously trained models and annotation efforts can be preserved while benefiting from the improved architecture and modular pipeline introduced in v2. In addition, we provide a migration pathway and accompanying documentation to assist users in transitioning existing workflows to the new system (Ratas, 2026).

3 Results

3.1 Overall NER+L performance

MedCAT v2 maintains equivalent performance to MedCAT v1 when using the same trained model and default vector context model for entity linking. Across all evaluated datasets, precision, recall, and F1-scores remain within the expected variance range using a private internal model[2]. To enable direct comparison, equivalent pipeline configurations and model parameters were used across both versions wherever possible. These results confirm that the redesigned modular architecture does not adversely affect core NER+L functionality. A detailed comparison between the two versions is shown in Table 1.

To evaluate performance parity between versions, we evaluated both systems across several biomedical NER+L datasets including MDACE (Cheng et al., 2023), COMETA (Basaldella et al., 2020), Distemist (Miranda-Escalada et al., 2022), the SNOMED Linking Challenge (Davidson et al., 2025), and MedMentions (Mohan and Li, 2019). All datasets were preprocessed into a common evaluation format in order to obtain comparable metrics.

3.2 Runtime performance

We compared MedCAT v1 and v2 across several common operational scenarios, including model loading, inference, and unsupervised training. Across all experiments, runtime performance remains broadly comparable between versions. Inference latency is effectively unchanged when equivalent configurations are used, indicating that the architectural refactor does not introduce additional overhead during normal pipeline execution. Model loading times show modest variation, with v2 slightly faster in some cases, while unsupervised training times remain within the expected range of implementation and hardware variability. A summary of timing benchmarks is presented in Table 2. 20 MIMIC-IV documents were used within the inference unsupervised training runs depicted in the table. All experiments were run on a single machine with an Apple M1 CPU and 32 GB RAM.

To ensure comparability, all timing experiments were performed under identical conditions and with the same base model configuration. The results therefore reflect differences introduced by architectural changes alone, rather than variations in model structure or parametrisation.

3.3 Performance–throughput trade-offs across configurations

A key advantage of the MedCAT v2 architecture is that it exposes multiple interchangeable components, allowing users to configure the pipeline to match their desired balance between predictive performance and processing speed. This flexibility is particularly important across the diverse range of MedCAT use cases, from large-scale processing of millions of EHRs across extensive concept vocabularies to targeted extraction tasks involving a small number of concepts.

[2]This model is trained on NHS data and therefore cannot be publicly released.

Table 1: Comparison of MedCAT v1 and MedCAT v2 performance across multiple biomedical NER+L datasets using a private internal model. Differences between versions are shown in the "diff." columns.

Dataset	Precision			Recall			F1		
	v1	v2	diff.	v1	v2	diff.	v1	v2	diff.
MDACE	0.3656	0.3660	0.0004	0.5391	0.5398	0.0007	0.4357	0.4362	0.0005
COMETA	0.9242	0.9245	0.0003	0.4521	0.4521	0.0000	0.6072	0.6072	0.0001
Distemist	0.3187	0.3192	0.0005	0.3949	0.3958	0.0009	0.3527	0.3534	0.0007
Linking Challenge	0.5353	0.5353	0.0000	0.3338	0.3337	-0.0001	0.4112	0.4112	0.0000
Med-Mentions	0.3521	0.3521	0.0000	0.5104	0.5103	0.0001	0.4167	0.4167	0.0000

Table 2: Runtime comparison between MedCAT v1 and v2 across inference, model loading, and unsupervised training scenarios. MC refers to MetaCAT components included in the pipeline.

Scenario	Ver	Model	Time - v1 (s)	Time - v2 (s)
inference	v1	NER (no MC)	9.090	10.827
inference	v1	NER + 3 MCs	22.537	22.311
load	v1	NER (no MC)	12.030	12.061
load	v1	NER + 3 MCs	12.218	17.653
unsup_train	v1	NER (no MC)	12.121	10.147
unsup_train	v1	NER + 3 MCs	22.392	26.776

On COMETA, the lightweight components deliver substantial runtime gains with only modest reductions in F1. For example, the combination of the regex tokenizer and the faster linker can achieve more than an order-of-magnitude speedup relative to the standard configuration, while maintaining comparable precision and only a small reduction in recall. This suggests that for some large-scale or latency-sensitive applications, MedCAT v2 can operate at significantly higher throughput without materially compromising output quality.

In contrast, results on the Linking Challenge dataset highlight that no single configuration is universally optimal. While the faster linker again provides clear speed improvements, the regex tokenizer does not produce a similar benefit. This likely reflects differences in dataset structure: COMETA consists of smaller, tightly filtered projects, whereas the Linking Challenge dataset allows a much larger variety of candidate concepts. As a result, the regex tokenizer produces many more potential mentions that must be evaluated during linking, increasing overall processing time. These results illustrate that tokenization and linking behaviour can interact with dataset characteristics in non-trivial ways.

The embedding-based linker represents the high-resource end of this configuration spectrum. While it is designed to improve disambiguation for short or ambiguous mentions, it introduces substantially higher computational cost and typically requires GPU acceleration for practical use. Across the evaluated datasets it does not consistently outperform lighter linking strategies, sometimes offering only marginal improvements despite significantly longer runtimes. This suggests that contextual embedding approaches are most beneficial in settings where ambiguity is a dominant challenge, but provide less advantage when candidate sets are already well constrained.

Overall, these experiments demonstrate that MedCAT v2 is intentionally flexible rather than prescriptive. Instead of enforcing a single configuration, the system allows practitioners to select pipeline components that match their operational constraints, whether prioritising throughput, resource efficiency, or predictive performance. The variation across datasets also highlights the importance of validating component choices on task-specific data rather than assuming a universally optimal configuration.

4 Discussion

The goal of this work was to address practical deployment and maintenance challenges encountered with MedCAT in real-world clinical and research

Table 3: Performance and runtime for different MedCAT v2 configurations across two datasets. Embedding linker results were obtained using the *abhinand/MedEmbed-small-v0.1* model

Dataset	Configuration	Precision	Recall	F1	Time (s)
COMETA					
Spacy	Vector context	0.9245	0.4521	**0.6072**	68.16
Spacy	Faster linker	**0.9266**	0.4225	0.5804	51.64
Spacy	Embedding linker	0.8871	**0.4455**	0.5932	321.37
Regex	Vector context	0.9130	0.4136	0.5693	30.54
Regex	Faster linker	0.9205	0.4108	0.5681	**6.21**
Regex	Embedding linker	0.8759	0.4394	0.5852	348.79
2023 Linking Challenge					
Spacy	Vector context	0.5353	0.3337	0.4112	75.40
Spacy	Faster linker	**0.5934**	0.2873	0.3871	**48.05**
Spacy	Embedding linker	0.5171	0.3557	**0.4215**	511.19
Regex	Vector context	0.4522	0.3162	0.3722	117.55
Regex	Faster linker	0.5091	0.2862	0.3664	82.61
Regex	Embedding linker	0.4141	**0.3593**	0.3847	248.02

environments, while preserving its established performance. Our results show that MedCAT v2 maintains equivalent predictive behaviour to earlier releases while substantially increasing flexibility through a modular, registry-based architecture.

Operational experience with MedCAT v1 highlighted recurring challenges: tightly coupled components made it difficult to experiment with alternative models, upgrades often required invasive code changes, and deployment across heterogeneous infrastructure introduced avoidable complexity. MedCAT v2 directly targets these issues by decoupling core pipeline elements and enabling components to be composed, replaced, or omitted without modification of the core library, and allowing new functionality to be integrated through external extensions. In practice, this allows development teams to integrate new tokenizers, linkers, or contextual classifiers independently, while clinical informatics groups can deploy lightweight CPU-only configurations tailored to local constraints. The extension system enabled external packages to provide components for MedCAT without cluttering up the core library.

The redesigned framework also exposes explicit trade-offs between accuracy, throughput, and resource consumption. Our experiments demonstrate that simplified components can yield substantial gains in processing speed, whereas embedding-based linking can improve disambiguation in some contexts but at substantially higher computational cost. These findings reinforce that no single configuration is optimal across deployments, and that teams should evaluate component choices against their own corpora, performance requirements, and infrastructure availability rather than relying solely on aggregate benchmark metrics.

Clinical NLP systems span a wide spectrum, from lightweight rule-based pipelines to fully neural end-to-end architectures. MedCAT v2 is designed to occupy a practical middle ground, supporting constrained environments while remaining extensible to more resource-intensive components where applicable. Rather than replacing specialised systems, the framework aims to lower the barrier to deployment and experimentation in settings where operational considerations often dominate model selection.

Several limitations remain. Embedding-based linking is computationally demanding and may be impractical without GPU access. Performance continues to depend on ontology coverage and data availability, with openly available models lagging institution-specific deployments. Increased configurability also introduces a larger design surface for users. To mitigate this, MedCAT v2 is accompanied by migration tooling, documentation, tutorials, and reference configurations, with ongoing refinement guided by community feedback.

Overall, MedCAT v2 provides a more sustainable engineering foundation for clinical concept extraction. By reducing operational friction while retaining established performance, the framework supports broader adoption of clinical NLP across

research and applied healthcare environments.

5 Conclusions

MedCAT v2 delivers a more operationally practical foundation for clinical named entity recognition and linking, preserving the established performance of earlier releases while substantially improving deployment flexibility and maintainability. By decoupling core components and simplifying integration, the framework reduces engineering overhead and enables teams to adapt pipelines to their local infrastructure and use-case requirements.

The redesigned architecture supports a wide range of deployment profiles, from lightweight CPU-only configurations within hospital infrastructure to more advanced pipelines that incorporate embedding-based components. The modular design also enables external extensions to provide alternative implementations of core components without requiring changes to the framework itself. Crucially, backward compatibility ensures continuity for existing users: previously trained models and data formats can be reused without modification, protecting prior investments while enabling incremental adoption of new functionality.

As an open-source platform, MedCAT v2 (Ratas et al., 2026) provides a sustainable basis for community-driven development and applied clinical NLP. The modular design positions the toolkit to integrate emerging methods in linking, contextualisation, and privacy-preserving workflows without disruptive architectural changes. By lowering barriers to experimentation, deployment, and long-term maintenance, MedCAT v2 supports more robust translation of NLP research into real-world healthcare environments and enables scalable extraction of structured clinical information from unstructured text.

References

Shubham Agarwal, Vlad Dinu, Thomas Searle, Mart Ratas, Anthony Shek, Dan F. Stein, James Teo, and Richard Dobson. 2025a. RelCAT: Advancing extraction of clinical inter-entity relationships from unstructured electronic health records. *Preprint*, arXiv:2501.16077.

Shubham Agarwal, Thomas Searle, Mart Ratas, Anthony Shek, James Teo, and Richard Dobson. 2025b. A framework for flexible extraction of clinical event contextual properties from electronic health records. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 6: Industry Track)*, pages 946–959, Vienna, Austria. Association for Computational Linguistics.

Joshua Au Yeung, Anthony Shek, Thomas Searle, Zeljko Kraljevic, Vlad Dinu, Mart Ratas, Mohammad Al-Agil, Aleksandra Foy, Barbara Rafferty, Vitaliy Oliynyk, and 1 others. 2024. Natural language processing data services for healthcare providers. *BMC medical informatics and decision making*, 24(1):356.

Marco Basaldella, Fangyu Liu, Ehsan Shareghi, and Nigel Collier. 2020. COMETA: A corpus for medical entity linking in the social media. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3122–3137, Online. Association for Computational Linguistics.

Daniel Capurro, Meliha Yetisgen, Erik Eaton, Robert Black, and Peter Tarczy-Hornoch. 2014. Availability of structured and unstructured clinical data for comparative effectiveness research and quality improvement: A multi-site assessment. *eGEMs (Generating Evidence & Methods to improve patient outcomes)*.

Hua Cheng, Rana Jafari, April Russell, Russell Klopfer, Edmond Lu, Benjamin Striner, and Matthew Gormley. 2023. MDACE: MIMIC documents annotated with code evidence. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7534–7550, Toronto, Canada. Association for Computational Linguistics.

Rory Davidson, Will Hardman, Guy Amit, Yonatan Bilu, Vincenzo Della Mea, Aleksandr Galaida, Irena Girshovitz, Mikhail Kulyabin, Mihai Horia Popescu, Kevin Roitero, Gleb Sokolov, and Chen Yanover. 2025. Snomed ct entity linking challenge. *Journal of the American Medical Informatics Association*, 32(9):1397–1406.

Luis B. Elvas, Ana Almeida, and João C. Ferreira. 2025. Natural language processing in medical text processing: A scoping literature review. *International Journal of Medical Informatics*, 204:106049.

Hadi Kharrazi, Laura J Anzaldi, Leilani Hernandez, Ashwini Davison, Cynthia M Boyd, Bruce Leff, Joe Kimura, and Jonathan P Weiner. 2018. The value of unstructured electronic health record data in geriatric syndrome case identification. *J. Am. Geriatr. Soc.*, 66(8):1499–1507.

Zeljko Kraljevic, Thomas Searle, Anthony Shek, Lukasz Roguski, Kawsar Noor, Daniel Bean, Aurelie Mascio, Leilei Zhu, Amos A. Folarin, Angus Roberts, Rebecca Bendayan, Mark P. Richardson, Robert Stewart, Anoop D. Shah, Wai Keong Wong, Zina Ibrahim, James T. Teo, and Richard J.B. Dobson. 2021. Multi-domain clinical natural language processing with MedCAT: The medical concept annotation toolkit. *Artificial Intelligence in Medicine*, 117:102083.

Leibo Liu, Victoria Blake, Matthew Barman, Blanca Gallego, Timothy Churches, Georgina Kennedy, Sze-Yuan Ooi, Geoffrey P Delaney, and Louisa Jorm. 2025. Using natural language processing to extract information from clinical text in electronic medical records for populating clinical registries: a systematic review. *Journal of the American Medical Informatics Association*, page ocaf176.

Aurelie Mascio, Zeljko Kraljevic, Daniel Bean, Richard Dobson, Robert Stewart, Rebecca Bendayan, and Angus Roberts. 2020a. Comparative analysis of text classification approaches in electronic health records. In *Proceedings of the 19th SIGBioMed Workshop on Biomedical Language Processing*, pages 86–94, Online. Association for Computational Linguistics.

Aurelie Mascio, Željko Kraljević, Daniel Bean, Richard Dobson, Robert Stewart, Rebecca Bendayan, and Angus Roberts. 2020b. Comparative analysis of text classification approaches in electronic health records. In *Proceedings of the 19th sigbiomed workshop on biomedical language processing*, pages 86–94.

Antonio Miranda-Escalada, Luis Gascó, Salvador Lima-López, Eulàlia Farré-Maduell, Darryl Estrada, Anastasios Nentidis, Anastasia Krithara, Georgios Katsimpras, Georgios Paliouras, and Martin Krallinger. 2022. Overview of distemist at bioasq: Automatic detection and normalization of diseases from clinical texts: results, methods, evaluation and multilingual resources. *CLEF (Working Notes)*, 3180:179–203.

Sunil Mohan and Donghui Li. 2019. Medmentions: A large biomedical corpus annotated with umls concepts. *arXiv preprint arXiv:1902.09476*.

Kawsar Noor, Lukasz Roguski, Xi Bai, Alex Handy, Roman Klapaukh, Amos Folarin, Luis Romao, Joshua Matteson, Nathan Lea, Leilei Zhu, and 1 others. 2022. Deployment of a free-text analytics platform at a uk national health service research hospital: Cogstack at university college london hospitals. *JMIR medical informatics*, 10(8):e38122.

Mart Ratas. 2026. MedCAT v2 migration guide. `https://github.com/CogStack/cogstack-nlp/blob/main/medcat-v2/docs/migration_guide_v2.md`. Accessed: 2026-04-24.

Mart Ratas, Zeljko Kraljevic, Anthony Shek, Thomas Searle, and Xi Bai. 2026. Medcat v2.

Tom M Seinen, Jan A Kors, Erik M van Mulligen, and Peter R Rijnbeek. 2025. Using structured codes and free-text notes to measure information complementarity in electronic health records: Feasibility and validation study. *J Med Internet Res*, 27:e66910.

Liesbet Van Bulck, Meghan Reading Turchioe, Maxim Topaz, and Jiyoun Song. 2024. Exploring the full potential of the electronic health record: the application of natural language processing for clinical practice. *European Journal of Cardiovascular Nursing*, 24(2):332–337.

Bram Van Es, Leon C Reteig, Sander C Tan, Marijn Schraagen, Myrthe M Hemker, Sebastiaan RS Arends, Miguel AR Rios, and Saskia Haitjema. 2023. Negation detection in dutch clinical texts: an evaluation of rule-based and machine learning methods. *BMC bioinformatics*, 24(1):10.

Honghan Wu, Minhong Wang, Jinge Wu, Farah Francis, Yun-Hsuan Chang, Alex Shavick, Hang Dong, Michael TC Poon, Natalie Fitzpatrick, Adam P Levine, and 1 others. 2022. A survey on clinical natural language processing in the united kingdom from 2007 to 2022. *NPJ digital medicine*, 5(1):186.

Urchade Zaratiana, Nadi Tomeh, Pierre Holat, and Thierry Charnois. 2024. GLiNER: Generalist model for named entity recognition using bidirectional transformer. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 5364–5376, Mexico City, Mexico. Association for Computational Linguistics.