A Survey of Big Data Approaches in Natural Language Formation and Computing Science

Aakash Kharb
Maharshi Dayanand University, Rohtak

DOI:10.37648/ijrst.v15i01.007

[1]*Received: 29 January 2025; Accepted: 25 February 2025; Published: 10 March 2025*

ABSTRACT

The convergence of big data, language technologies and computing science has reshaped how we model human languages and software systems. Massive text corpora, multimodal datasets and "big code" repositories enable data-driven accounts of language formation and change, while also powering advances in software analytics, code intelligence and large language models (LLMs) for natural and programming languages. This paper surveys and synthesizes applications of big data across two tightly coupled domains: (1) language formation in the sense of acquisition, usage and diachronic change, and (2) computing science, particularly natural language processing (NLP), LLMs and code-centric analytics. We review foundational big-data NLP and corpus-linguistic work, recent LLM surveys, and research on mining software repositories and big code. We then propose a conceptual framework connecting human language corpora and software repositories as parallel manifestations of "languages in use," both amenable to large-scale statistical modelling. Comparative analysis contrasts traditional small-data, rule-based methodologies with big-data, representation-learning approaches across linguistic and software-engineering tasks. Case studies include social-media NLP, corpus-based language change modelling, AI-assisted programming and software analytics pipelines. Finally, we discuss open challenges, including data bias, interpretability, governance, privacy and the risk of over-fitting linguistic and programming norms to dominant platforms.

Keywords: *Big data; language formation; corpus linguistics; large language models; big code; mining software repositories; software analytics; natural language processing.*

1. Introduction

The last decade has seen an unprecedented growth in the volume, variety and velocity of language-related data: web pages, social media, scientific literature, voice transcripts and code repositories. Big data technologies make it feasible to store and analyse petabytes of text and source code, and deep learning architectures can exploit these resources to build powerful language models.

In linguistics, this scale enables corpus-based exploration of language formation, acquisition, and change, going beyond small, curated datasets to "big and rich" corpora that capture real-world usage across genres and time. In computing science, similar shifts have occurred: large software repositories and "big code" corpora are mined to learn patterns of software development, support automated code completion and detect bugs.

[1] How to cite the article: Kharb A.; March 2025; A survey of Big Data Approaches in Natural Language Formation and Computing Science; *International Journal of Research in Science and Technology*, Vol 15, Issue 1, 85-93, DOI: http://doi.org/10.37648/ijrst.v15i01.007

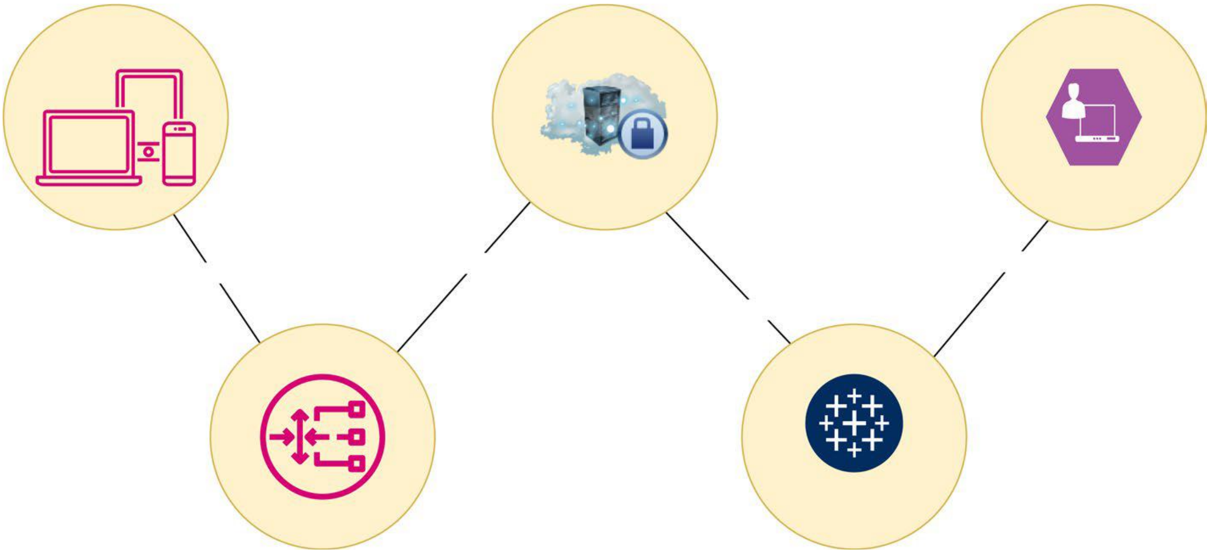

Fig. 1: Structure of the big data cycle

2. Literature Review

2.1 Big data and natural language processing

Wang et al. framed "natural language processing systems and big data analytics" as a core combination for scalable text understanding, highlighting distributed storage, parallel processing and statistical NLP. Hirschberg and Manning reviewed advances in NLP, emphasizing how large, heterogeneous corpora and deep learning transformed tasks like parsing, machine translation and information extraction.

Subsequent work explored the tight coupling between big data infrastructures and NLP pipelines, using Hadoop/Spark, NoSQL stores and streaming architectures for large-scale sentiment analysis, topic modeling and entity extraction on social media and logs.

2.2 Corpus linguistics, language change and language formation

Big data has reshaped corpus linguistics. Nevalainen demonstrated how "large recent corpora" enable quantitative study of language change, e.g., tracking lexical replacements or grammatical shifts over centuries and comparing incipient vs completed changes. Recent introductions to corpus linguistics in the era of "big and rich data" emphasize methodological innovations for languages like Spanish and Portuguese, where large web-scale corpora and rich metadata support sociolinguistic and diachronic analyses.

This perspective interprets language formation not as a static rule system but as a dynamic distribution over usage patterns, shaped by social, cognitive and technological forces.

2.3 LLMs and big-data language modelling

Surveys of large language models show how scaling data and parameters enables emergent capabilities in reasoning, dialogue and multi-task generalization. These models are trained on diverse web corpora, code, and domain-specific texts, blurring boundaries between "general" language and specialized registers.

Chang et al. review evaluation methodologies for LLMs, noting that benchmarking must consider both traditional NLP tasks and broader social impacts such as safety and bias.

2.4 Big code, software analytics and mining software repositories

Allamanis et al. survey machine learning for "big code and naturalness," arguing that source code exhibits statistical regularities similar to natural language and can be modeled with n-grams, neural sequence models and graph-based representations. Wong et al. review natural-language generation and understanding of big code in AI-assisted programming, focusing on transformer-based models for code generation, summarization and defect detection.

Mining Software Repositories (MSR) is another strand of big-data computing science. Vidoni proposes a systematic process for MSR, from repository selection to data extraction, cleaning and analysis. Gousios demonstrates big-data software analytics using Apache Spark on large event streams from development platforms.

Table 1. Main strands in the literature

Strand	Representative works	Big-data artefact
Big-data NLP	Wang et al. 2015; Hirschberg & Manning 2015	Web/social media corpora, logs
Corpus linguistics & change	Nevalainen 2020; corpus-rich introductions	Historical and contemporary text corpora
LLMs & evaluation	Kumar 2024; Chang et al. 2024	Web-scale multilingual corpora, code, docs
Big code & AI-assisted dev	Allamanis et al. 2018; Wong et al. 2023; Wan et al. 2024	Code repositories, issue trackers, Q&A sites
MSR & software analytics	Vidoni 2022; Gousios 2018	Version control, CI logs, bug trackers

3. Big data and language formation

3.1 From grammar to distributions over usage

Big data reframes language formation as a distribution over observed usage rather than a fixed rule system. Large corpora allow detailed estimation of frequency, collocation, syntactic preferences and lexical innovation, which in turn inform theories of how languages stabilize and change.

For instance, diachronic corpora can trace the rise and fall of competing constructions (e.g., modal verbs, progressive forms), while social media data reveals real-time lexical innovation and borrowing. When combined with demographic metadata, such corpora also shed light on how age, region and social networks influence language formation.

3.2 Studying language change with large corpora

Nevalainen shows that corpora comprising hundreds of millions of words enable fine-grained periodization of language change, helping distinguish “incipient” from “completed” changes and modeling their trajectories. Contemporary corpus projects extend this logic with multi-genre, multilingual datasets and dynamic visualizations of change.

Big data also supports semantic change detection, where vector representations trained on different time slices allow comparison of word meanings across decades.

3.3 Big data in language learning and pedagogy

Large learner corpora and data-driven learning (DDL) approaches expose language learners to authentic usage patterns and support personalized feedback. Systematic reviews report increasing use of corpora in language education, allowing teachers to design tasks around concordances and usage statistics rather than intuition alone.

Table 2. Applications of big data to language formation

Application area	Big-data source	Example benefit
Diachronic change	Historical and web corpora	Modeling trajectories of grammatical change
Socio-linguistics	Social media, demographic metadata	Linking variants to age, region, social network
Learner language	Learner corpora, classroom data	Data-driven feedback and material design
Semantic change	Time-sliced corpora, embeddings	Detecting emerging senses and discarding old ones

Comparatively, small hand-picked corpora allowed qualitative insights but lacked coverage; big data enables statistically robust claims but raises new questions about representativeness and bias.

4. Big data in computing science

4.1 Big-data NLP systems

Big-data NLP systems integrate distributed storage (e.g., HDFS, NoSQL) with scalable processing frameworks (e.g., MapReduce, Spark) to handle text analytics over billions of documents. Wang et al. outline architectures where raw text flows through pipelines including tokenization, parsing, feature extraction and machine learning models, all parallelized across clusters.

Such systems support applications like sentiment analysis over social media streams, large-scale information extraction from scientific literature, and monitoring of customer feedback.

4.2 LLMs as big-data learners

Modern LLMs are trained on trillions of tokens spanning natural language, code and multimodal content. Surveys highlight how scaling both data and parameters is central to their performance, with transformer architectures learning contextual embeddings that serve as universal representations.

Kumar’s 2024 survey emphasizes that LLM pipelines depend heavily on big-data engineering: deduplication, filtering, domain balancing and continual updates, alongside specialized hardware and distributed training algorithms.

4.3 Code intelligence and big code

In code intelligence, big data appears as massive codebases (e.g., GitHub) combined with issue trackers, pull requests and Q&A forums. Allamanis et al. show that probabilistic and neural models trained on these corpora can support code completion, naming, bug detection and clone detection.

Wong et al. review “big code” LLMs such as Codex and AlphaCode, which treat code as another language and can generate, translate and explain programs. Wan et al. extend this with a deep-learning-for-code-intelligence toolkit and benchmarks that standardize evaluation on large corpora.

4.4 Mining software repositories and software analytics

MSR research treats software repositories as big-data sources to study developer behaviour, project evolution and defect dynamics. Vidoni’s systematic review proposes guidelines for selecting repositories, cleaning data and designing analyses that are statistically sound. Gousios’s work on Spark-based analytics shows how commit events, issues and code diffs can be processed in near real time.

Table 3. Big-data artefacts in computing science

Artefact type	Examples	Typical tasks
Text corpora	Web pages, news, social media	NLP tasks, LLM pre-training
Big code repositories	GitHub, GitLab, package registries	Code completion, repair, summarization
Software repositories	Version control, issue trackers, CI logs	MSR, defect prediction, process analytics
Scientific literature	Digital libraries, preprint servers	Knowledge extraction, citation analysis

5. Conceptual framework: bridging language formation and computing science

Although linguistics and computing science appear distinct, big data exposes deep parallels: both natural language and code are symbolic systems governed by implicit constraints and social conventions, observable through large-scale usage data.

We can conceptualize an integrated pipeline with four stages:

1. **Data acquisition & curation** – collecting corpora or repositories, cleaning, annotating and ensuring governance.
2. **Representation learning** – learning embeddings or structured representations (e.g., syntax trees, graphs).
3. **Pattern discovery & modelling** – inferring regularities such as syntactic preferences, coding idioms or bug patterns.
4. **Application & feedback** – deploying models in tools (e.g., educational platforms, IDE assistants) and using feedback to refine both models and data.

This framework is agnostic to whether tokens are words or program statements; the underlying big-data methods are shared.

Table 4. Unified big-data framework for language and code

Stage	Natural language example	Computing science example
Acquisition	Web-scale English corpus	Multi-project Java/GitHub repository
Representation	Contextual word embeddings, parse trees	ASTs, control-flow graphs, code embeddings
Pattern discovery	Collocations, syntactic alternations, semantic shift	API usage patterns, defect-prone components
Application	LLM-based writing assistant, language tutor	Code completion, refactoring suggestions

Comparatively, earlier small-scale studies could span only one or two stages; big data allows continuous, end-to-end loops where model outputs also influence future data (e.g., through tool adoption).

6. Applications and case studies

6.1 Social-media NLP and opinion mining

Big-data NLP has been widely applied to social media for sentiment analysis, stance detection and topic modeling. Reviews on the integration of big-data analytics and NLP describe architectures where high-throughput ingestion (e.g., Twitter firehose) feeds streaming sentiment models and dashboards for brands or policy-makers.

Comparatively, pre-big-data sentiment analyses relied on small survey-like samples and manual annotation; modern systems can incorporate millions of multilingual posts per hour.

6.2 Clinical and domain-specific NLP

Systematic reviews of radiology NLP and related domains show that large corpora of clinical reports enable automated coding, cohort identification and outcome prediction, though performance and generalizability vary across institutions. These are quintessential big-data applications, requiring robust de-identification, terminology mapping and domain-specific language models.

6.3 Corpus-based language education

In education, DDL systems enable learners to search concordances from large corpora, examine authentic examples and derive rules inductively. Big data supports adaptive feedback, where learner errors are matched against common patterns and remedial materials are recommended.

6.4 AI-assisted programming tools

AI-assisted coding tools like Copilot rely on big-code LLMs that exploit learned distributions over code tokens to suggest completions, generate tests and refactor code. Wong et al. document how these models are trained on huge codebases and demonstrate performance gains over traditional, pattern-based tools.

Wan et al. provide benchmarks and toolkits that standardize how such models are evaluated on tasks like code summarization and bug detection, emphasizing the importance of large, diverse datasets.

6.5 Software analytics for process improvement

Gousios's Spark-based analytics and Vidoni's MSR process illustrate how version control and issue data can be mined to detect hotspots, estimate refactoring opportunities and anticipate build failures. Organizations use these insights to guide code reviews, resource allocation and architectural decisions.

Table 5. Representative big-data applications

Domain	Big-data input	Output / tool type
Social media	Posts, comments, reactions	Sentiment dashboards, trend detection
Healthcare	Clinical reports, EHR text	Automated coding, cohort discovery
Education	Learner corpora, usage logs	Adaptive grammar feedback, corpus-based tasks
Programming	Big code + issues/PRs	Code assistants, automated testing, refactoring
Software process	Commits, build logs, bug reports	Risk analytics, productivity metrics

7. Comparative analysis

7.1 Traditional vs big-data approaches in language studies

Traditional linguistic analysis, especially in historical and theoretical work, relied on small collections of curated examples, sometimes augmented by modest corpora. Big-data corpus linguistics offers coverage, statistical power and the ability to test competing hypotheses over millions of tokens. However, big data may over-represent high-resource languages, online genres and specific socio-economic groups.

7.2 Rule-based vs data-driven NLP

Rule-based NLP systems encode linguistic knowledge explicitly, which aids interpretability but struggles with robustness and domain transfer. Big-data, data-driven approaches learn patterns directly from corpora, accommodating ambiguity and variation but often sacrificing transparency. Hirschberg and Manning highlight how deep learning, driven by big data, matched or exceeded rule-based systems across core tasks.

7.3 Language vs code as big-data targets

Natural language and code share properties (discrete tokens, hierarchical structures, statistical regularities) but differ in ambiguity and correctness constraints. Allamanis et al. argue that code is more regular and tightly constrained, making it easier to model in some respects, while also requiring exact syntactic correctness. In contrast, natural language modeling must contend with polysemy, figurative language and open-class vocabularies.

Table 6. Comparative analysis across domains

Dimension	**Traditional linguistics / SE**	**Big-data language & code modeling**
Data scale	Thousands to millions of tokens	Billions to trillions of tokens
Methodology	Manual, rule-based, small-sample	Statistical, deep learning, representation learning
Interpretability	High (explicit rules)	Lower (complex models), requires explainability
Coverage	Limited genres/languages	Wide but skewed toward high-resource domains
Error tolerance	Qualitative insight focus	Quantitative metrics, error-aware deployment

7.4 Evaluation and benchmarking

The evaluation of big-data models has itself become a research topic. Chang et al. stress that LLM evaluation requires diverse benchmarks beyond accuracy, including robustness, fairness and downstream impact. Wan et al. and Wong et al. analogously propose standardized benchmarks for code intelligence to ensure reproducible comparisons across models and datasets.

8. Challenges and future directions

Despite impressive progress, integrating big data into language formation studies and computing science raises several challenges.

1. **Data quality and bias.** Web and repository data are noisy and skewed. Over-representation of certain languages, dialects, frameworks or organizations can lead to biased models and misleading inferences.
2. **Interpretability and trust.** LLMs and deep code models operate as black boxes. For both educational and safety-critical domains (e.g., healthcare, finance), interpretable models or post-hoc explanations are important.

3. **Ethics, privacy and governance.** Clinical text, private communication and proprietary code have strong privacy constraints; governance frameworks and anonymization pipelines are essential.
4. **Feedback loops and norm shaping.** As LLMs and code assistants become widely used, they may reshape linguistic and programming norms (e.g., promoting certain idioms or styles), which then feed back into new training data.
5. **Resource disparities.** High-resource institutions and languages benefit most from big-data infrastructure; low-resource languages and small organizations may lag.

Future directions include:

- Cross-lingual and cross-modal corpora for more inclusive language formation studies.
- Joint modeling of natural language and code, supporting end-to-end pipelines from requirements to implementation.
- Integration of symbolic constraints with big-data models to improve safety and controllability.
- Community benchmarks for language-formation tasks (e.g., semantic change, acquisition patterns) akin to MSR and code-intelligence benchmarks.

Table 7. Key challenges and research opportunities

Challenge area	Specific issue	Opportunity
Bias & coverage	Over-represented languages/frameworks	Curated, balanced datasets; debiasing methods
Interpretability	Black-box LLMs and code models	Explainable AI, hybrid symbolic–neural models
Governance	Privacy of clinical text and proprietary code	Federated learning, privacy-preserving analytics
Feedback loops	Models reshaping language/code norms	Longitudinal measurement of norm changes

9. Conclusion

Big data has transformed both the study of language formation and the practice of computing science. Large, diverse corpora enable linguists to quantify change, variation and acquisition at scales previously impossible, while big-code and MSR datasets allow computing researchers to model software development processes and build AI-assisted tools.

Despite domain differences, both natural and programming languages can be understood as evolving systems, whose regularities become visible only at scale. Big data, combined with deep learning, underpins modern LLMs that bridge these domains by jointly modeling text and code. Comparative analysis reveals that big-data approaches offer coverage and performance gains but raise challenges around bias, interpretability and governance.

For researchers and practitioners, the path forward lies in combining the strengths of traditional linguistic and software-engineering theories with the empirical power of big data, designing pipelines and evaluation frameworks that are not only accurate but also transparent, fair and aligned with human values.

References

1. Allamanis, M., Barr, E. T., Devanbu, P., & Sutton, C. (2018). A survey of machine learning for big code and naturalness. *ACM Computing Surveys*, 51(4), Article 81. https://doi.org/10.1145/3212695

2. Chang, Y., Wang, X., Wang, J., Wu, Y., Yang, L., Zhu, K., … Xie, X. (2024). A survey on evaluation of large language models. *ACM Transactions on Intelligent Systems and Technology*, 15(3), Article 39. https://doi.org/10.1145/3641289

3. Gousios, G. (2018). Big data software analytics with Apache Spark. In *Proceedings of the 40th International Conference on Software Engineering: Companion Proceedings* (pp. 542–543). ACM. https://doi.org/10.1145/3183440.3183458

4. Hirschberg, J., & Manning, C. D. (2015). Advances in natural language processing. *Science*, 349(6245), 261–266. https://doi.org/10.1126/science.aaa8685

5. Kumar, P. (2024). Large language models (LLMs): Survey, technical frameworks, and future challenges. *Artificial Intelligence Review*, 57(10), 260. https://doi.org/10.1007/s10462-024-10888-y

6. Nevalainen, T. (2020). Using large recent corpora to study language change. In R. D. Janda, B. D. Joseph, & B. Vance (Eds.), *Handbook of Historical Linguistics* (Vol. 2, pp. 272–290). Wiley. https://doi.org/10.1002/9781118732168.ch13

7. Theodorakopoulos, L., Antonopoulou, H., Halkiopoulos, C., & Mamalougkou, V. (2023). Synergizing big data analytics and natural language processing: A comprehensive review of techniques and emerging trends. *International Journal of Multidisciplinary and Current Educational Research*, 5(6), 111–118. (DOI as reported in the article: e.g., 10.28991/esj-2023-07-03-04 for related works)

8. Vidoni, M. (2022). A systematic process for mining software repositories: Results from a systematic literature review. *Information and Software Technology*, 144, 106791. https://doi.org/10.1016/j.infsof.2021.106791

9. Wang, L., Wang, G., & Alexander, C. A. (2015). Natural language processing systems and big data analytics. *International Journal of Computational Systems Engineering*, 2(2), 76–84. https://doi.org/10.1504/IJCSYSE.2015.077052

10. Wan, Y., He, Y., Bi, Z., Zhang, J., Zhang, H., Sui, Y., … Yu, P. S. (2024). Deep learning for code intelligence: Survey, benchmark and toolkit. *ACM Computing Surveys*, 56(12), 1–41. https://doi.org/10.1145/3664597

11. Wong, M.-F., Guo, S., Hang, C.-N., Ho, S.-W., & Tan, C.-W. (2023). Natural language generation and understanding of big code for AI-assisted programming: A review. *Entropy*, 25(6), 888. https://doi.org/10.3390/e25060888

12. Yayah, F. C., Ghauth, K. I., & Ting, C.-Y. (2018). Application of NLP on big data using Hadoop: Case study using trouble tickets. *Advanced Science Letters*, 24(10), 7696–7702. https://doi.org/10.1166/asl.2018.13002

13. Zeroual, I., Lakhouaja, A., & Belkredim, F. Z. (2018). Data science in light of natural language processing. *Procedia Computer Science*, 127, 58–67. https://doi.org/10.1016/j.procs.2018.01.099