



Language as action and coordination: A linguistic framework for agentic AI

Shu-Kai Hsieh

(National Taiwan University)

Hsieh, Shu-Kai. 2026. Language as action and coordination: A linguistic framework for agentic AI. *Linguistic Research* 43(Special Edition): 323-340. Recent advances in large language models, multimodal systems, and agentic AI invite a re-examination of language from the perspective of linguistic theory. This article argues that language in contemporary AI systems should be understood not only as a medium of representation, but also as an interface for action, a substrate for memory, and a coordination protocol for distributed intelligence. The argument proceeds in three steps. First, I reinterpret prompting and tool use through speech act theory and pragmatics: prompts, system instructions, and function calls do not merely describe desired outputs; they perform operations that reshape model behavior and sometimes trigger consequential actions. Second, I show that agentic AI extends this language-as-action view into a broader ecology of semi-structured communication, where natural language, JSON, Markdown, memory files, skill descriptions, handoff protocols, and governance rules jointly organize the work of agents. Third, I connect this framework to representational geometry and typology-aware evaluation, arguing that typological diversity provides a principled way to probe the structure and limits of model-internal semantic spaces. The resulting view treats LLMs and agent systems as both engineering artifacts and experimental environments for linguistic theory. It suggests that linguistics can contribute directly to AI by clarifying how meaning, context, action, repair, and social coordination are encoded, operationalized, and evaluated in artificial systems. (National Taiwan University)

Keywords agentic AI, computational pragmatics, language engineering, semantic geometry, linguistic typology

1. Introduction

For much of modern linguistic theory, language has been analyzed as a symbolic system: a structured set of units, rules, meanings, and usage conventions (Saussure 1916; Koerner 2013). This view remains indispensable. Yet the rapid development of large language models (LLMs), multimodal systems, and agentic AI has made it necessary to ask a further question: what happens when language becomes not only an object of analysis, but also a computational interface through which artificial systems are instructed, coordinated, constrained, and made to act?

This question is not merely terminological. In current agentic AI systems, language is no longer limited to user-facing dialogue. It appears in system prompts, tool schemas, function descriptions, memory records, skill files, planning traces, review rubrics, handoff protocols, safety policies, and governance rules. A prompt can change an output distribution; a tool call can retrieve a document, send a message, or modify a file; a memory note can shape future interactions; a governance instruction can block, redirect, or escalate an action. In short, language increasingly functions as operational infrastructure.

The central claim of this article is that agentic AI forces us to expand the linguistic horizon. Language should be understood along at least four interrelated dimensions. It is a representational system, as classical linguistics has long emphasized. It is also a pragmatic act, because prompts and tool calls perform operations. It is a geometric structure, because meanings and relations are encoded in model-internal representational spaces. Finally, it is a coordination protocol, because multi-agent systems depend on role definitions, task decomposition, handoff procedures, memory management, critique, repair, and governance.

This article therefore does not argue that LLMs “understand” language in the same way humans do (Bender and Koller 2020). Nor does it claim that neural models replace traditional linguistic theory (cf. Piantadosi 2023). Rather, it proposes a more modest and, I believe, more productive position: LLMs and agent systems provide a new experimental setting in which long-standing linguistic concepts, such as meaning, context, speech act, implicature, deixis, repair, genre, typology, and discourse organization, etc., can be operationalized and tested in computational form.

The article is organized as follows. Section 2 examines how LLMs reconstruct language as tokenized units, distributional meaning, attention-mediated context, and

aligned interaction. Section 3 reframes prompting and tool use as pragmatic action. Section 4 develops the notion of agentic orchestration, focusing on natural language, JSON, Markdown, memory, skills, and handoff protocols as complementary linguistic layers. Section 5 connects agentic language engineering with semantic geometry and typology-aware evaluation. Section 6 concludes by outlining a research agenda for linguistics in the age of agentic AI.

2. Language engineering in large language models

LLM training begins with a computational reconstruction of language. Before a model can generate text, language must be transformed into computable units, embedded into vectors, related through attention, and adjusted through post-training. Each stage carries an implicit linguistic theory. This theory is not usually stated in the vocabulary of linguistics, but it has direct consequences for how artificial systems represent and use language.

Tokenization turns written language into computable fragments. These fragments are not necessarily words in the traditional linguistic sense. They may be characters, subword units, recurrent strings, byte-level units, or combinations of these. Thus, the model's first encounter with language is already mediated by an engineering decision about segmentation. Linguistically, it is worth noting that first, although existing mainstream tokenization methods like Byte-Pair Encoding (BPE) originate from the field of data compression, researchers should stop designing tokenizers merely as standard data compressors. Instead, tokenization should be viewed as a crucial inductive bias where structural constraints (preserving word and digit boundaries) matter much more than achieving the shortest possible sequence length (Schmid et al. 2024). Second, this matters because tokenization is not linguistically-culturally neutral: languages with different writing systems, morphological structures, and orthographic conventions may be segmented with unequal efficiency, thereby affecting cost, context length, and representational access (Petrov et al. 2023).

Embeddings transform linguistic units into relational representations. In this setting, meaning is not a dictionary definition stored with a symbol, but a position in a high-dimensional space derived from patterns of use (Mikolov et al. 2013). This view resonates with distributional and usage-based approaches to meaning: linguistic

units are understood through their contextual behavior. In multimodal systems, this relational structure extends beyond text. Meaning becomes inter-semiotic, connecting words with images, sounds, gestures, and possible actions (Zhang et al. 2024).

Self-attention operationalizes context as dynamic relational weighting (Vaswani et al. 2017). Traditional linguistics treats context as a rich combination of co-text, speaker assumptions, discourse history, situation, and cultural background. Attention does not reproduce this full human notion of context, but it gives AI systems a computable mechanism for selectively reactivating relevant parts of an input. Long-distance dependencies, discourse topics, prior commitments, and local constraints can all, in principle, be mediated through attention-like mechanisms.

Post-training and alignment then move the system from grammatical or semantic fluency toward interactional appropriateness. Supervised fine-tuning, reinforcement learning from human feedback, constitutional training, and related methods teach models not only what can be said, but what should be said under a particular social, task, or safety regime. From a linguistic perspective, this is a movement from syntax and semantics toward pragmatics and discourse regulation.

In summary, language in LLMs is tokenized as units, embedded as meaning, attended to as context, and aligned as interaction. Once these layers are engineered, language becomes not merely a medium of expression but a computational infrastructure for thinking and coordination.

3. Prompting and tool use as pragmatic action

The shift from text generation to agentic behavior makes pragmatics central. In ordinary conversation, an utterance does not merely encode propositional content. It can request, promise, warn, authorize, apologize, refuse, or declare. This insight is the core of speech act theory (Austin 1962; Searle 1969). In AI systems, the same insight becomes newly literal: certain utterances, once interpreted by a model and routed through tools, can directly change computational or physical states.

A *prompt* is therefore not simply a sentence. It is a linguistic intervention that conditions a model's behavior. In minimal machine-learning terms, prompting is the operation of conditioning a model's output distribution on an input sequence (Brown et al. 2020). From a pragmatic perspective, however, this definition is too thin. A

prompt also carries illocutionary force: it asks the model to adopt a role, follow a constraint, perform a task, use a tool, refuse unsafe content, or produce a particular genre of response.

This view helps explain why prompt engineering is not merely a matter of wording. It involves the explicit design of conditions that humans often leave implicit. Human interlocutors routinely rely on shared background, genre expectations, deictic cues, and repair mechanisms (Grice 1975; Levinson 1983). AI agents, by contrast, may treat underspecified language as executable instruction. A request such as “send the revised file to the usual person” presupposes a shared understanding of “the revised file” and “the usual person,” etc. If these presuppositions are not recoverable from context or memory, an agent should initiate repair rather than act.

Tool use makes the pragmatic stakes even clearer. To overcome the inherent limitations of standard language models, such as hallucinations and the lack of access to real-time information, recent studies have augmented LLMs with external capabilities via tool learning (Qu et al. 2025). Pioneering frameworks like ReAct (Yao et al. 2023) and Toolformer (Schick et al. 2024) have demonstrated that LLMs can autonomously plan, decide when to invoke external APIs, and interpret the returned execution feedback to solve complex downstream tasks. Looking further, a function schema may specify the syntax of a call, and a docstring may describe its semantic content, but the crucial question is pragmatic: when should the tool be used, what side effects may follow, and when should the agent ask for confirmation? A search function, a delete function, and a payment function may all appear as tool calls in a system, but they differ radically in reversibility, risk, and required authorization. Many agent failures occur not at the level of syntax or even semantics, but at the level of pragmatic judgment.

This suggests a useful distinction. The schema of a tool is its *syntax*; the docstring is its *semantics*; and the invocation condition is its *pragmatics*. Agent design becomes safer and more reliable when these three layers are made explicit. A well-designed agent should know not only how to call a tool and what the tool means, but also whether the current utterance licenses the call, whether relevant presuppositions are satisfied, whether the action is reversible, and whether human approval is required.

4. Agentic orchestration as language-mediated coordination

Agentic AI extends the pragmatic view of prompting into a broader ecology of linguistic coordination. While primitive chatbot AI systems rely on static prompt chains or basic tool use, scaling to long-horizon, complex tasks requires a transition to agentic orchestration (Ruan et al. 2026). Under this paradigm, a centralized cognitive layer acts as an orchestrator that dynamically decomposes goals, manages collective memory, and spawns specialized sub-agents to execute sub-tasks. In this scenario, language organizes planning, tool use, memory, verification, delegation, and governance. Agent engineering can therefore be understood as a form of language engineering: the design of linguistic artifacts that allow computational components to coordinate action over time.

To be precise, this coordination does not rely on natural language alone. In practice, agent systems depend on several complementary structural representations. Natural language provides semantic richness, interpretation, and negotiation. JSON provides machine-readable structure for state, tools, and data exchange. Markdown provides a human-readable workspace for plans, reports, memory, review, and documentation. The three forms, as shown in Table 1, differ in ambiguity, readability, and machine verifiability, but they jointly support agentic work, whose components are introduced below.

Table 1. Three linguistic layers in agentic AI

Layer	Primary function	Typical forms	Main risk
Natural language	Meaning, interpretation, negotiation, explanation	Prompts, role instructions, critique, reflection, user requests	Ambiguity, underspecification, pragmatic misfire
JSON / structured data	State tracking, tool invocation, verification	Function calls, API payloads, memory records, evaluation results	Over-rigid schemas, loss of context, false precision
Markdown / semi-structured discourse	Shared workspace for humans and agents	Plans, reports, task boards, skill files, review notes	Stale instructions, conflicting memories, uncontrolled accumulation

4.1 Persona, memory, and skills

Agent persona files define who is serving whom, in what role, and under what interactional principles. A user profile specifies the addressee and collaboration context; an identity file specifies the agent's operational role; a value or "soul" file specifies preferred norms, boundaries, and interactional style. These files are not merely decorative. They function as durable speech acts that establish roles, obligations, and expectations across sessions.

Memory files define how an agent preserves context. They convert interactional history into reusable, compressed, and inspectable records. This is linguistically significant because memory is not simply storage. It is selection, summarization, framing, and forgetting. A memory entry may stabilize a preference, carry forward a project decision, or contaminate future reasoning if an unverified hypothesis is stored as fact. Memory management is therefore a discourse problem as well as an engineering problem.

Skill files define procedural competence. A skill is not simply a tool; it is a language-mediated routine that specifies when a capability should be invoked, what steps should be followed, and how the result should be checked. Skill invocation is consequently a pragmatic problem: the system must map an utterance to an intention, then map that intention to an appropriate procedure. This process resembles the movement from utterance to illocutionary force in human communication.

4.2 Handoff, critique, and repair

Multi-agent systems make coordination visible. A planner may decompose a task, an executor may use tools, a verifier may check results, a memory manager may update state, and a domain expert may supply specialized knowledge. Such systems succeed only when language specifies who is responsible for what, what counts as completion, what information must be passed forward, and how disagreements are resolved.

Handoff protocols are especially important. A vague handoff such as "continue from here" may be intelligible to a human colleague in a shared office, but it is often insufficient for an agent. A robust handoff should include the task goal, current state, completed steps, unresolved issues, assumptions, evidence, tool outputs, and

recommended next action. In this sense, the handoff document is a genre: a conventionalized linguistic form designed to preserve coordination across distributed actors (on common ground and joint action, Clark 1996).

Critique and repair are equally linguistic. In human conversation, repair mechanisms allow speakers to correct misunderstanding, request clarification, and revise commitments (Sacks et al. 1974; Schegloff et al. 1977). In agent systems, repair must often be engineered explicitly. The system should know when to ask a clarifying question, when to stop an unsafe action, when to escalate to a human, and when to revise a plan after tool feedback. Without such mechanisms, agents may transform minor ambiguities into consequential errors.

4.3 Governance as a linguistic layer

Governance files and policies define what an agent is permitted to do. They specify tool permissions, approval thresholds, risk categories, logging requirements, and escalation procedures. From a linguistic perspective, such governance documents are not peripheral. They are a normative layer of language that constrains action. They tell agents not merely what they can do, but what they may do, must do, and must not do.

This distinction clarifies the relationship between skills, tools, and governance. A skill defines “how to do something”; a tool defines “what can be used”; governance defines “under what conditions action is authorized.” Confusing these layers leads to fragile systems. A capable agent with poorly specified governance may act too freely; a heavily governed agent without clear skills may refuse or stall; an agent with tools but no pragmatic invocation rules may use the right function at the wrong time.

Agentic orchestration is thus best understood as a **layered linguistic system**: role language defines institutional identity, task language decomposes goals, handoff language coordinates work, critique language enables evaluation, memory language sustains continuity, and governance language regulates action. These layers can be viewed from two complementary angles. Figure 1 presents the functional layers that an agent coordinates through language, with language itself as the semiotic mediating layer at the center. Figure 2 situates the same activity within the nested scopes of agent engineering: from the innermost single utterance (prompt engineering) outward

through the assembly of context (context engineering), the packaging of reusable procedures (skill engineering), the wiring of tools, permissions, and control flow (harness engineering), and finally the multi-turn process in which behavior is evaluated and repaired (loop engineering). Each scope operates on the same medium - a flow of language - so that designing an agent is, at every level, a form of language engineering.

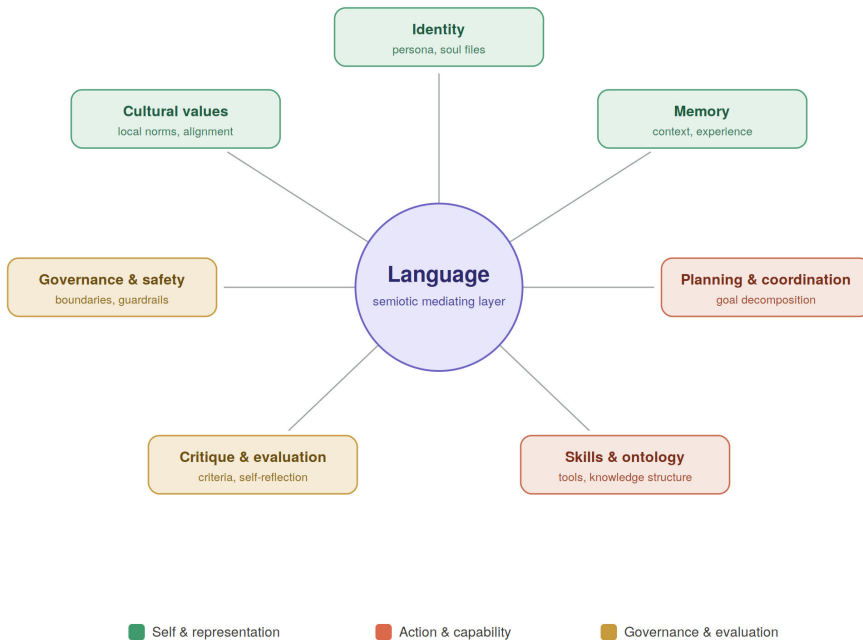


Figure 1. Language as the semiotic mediating layer of an agent system. Seven functional layers - identity, memory, cultural values, planning and coordination, skills and ontology, critique and evaluation, and governance and safety - are coordinated through language, grouped into three strata: Self and representation, action and capability, and governance and evaluation

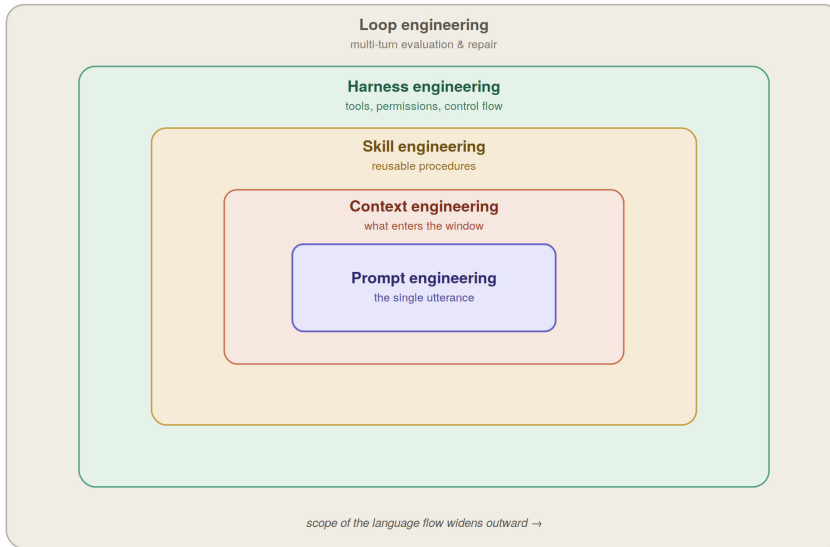


Figure 2. Agent engineering as nested linguistic scopes. From the innermost single utterance (prompt engineering) outward: Context engineering assembles what enters the window; skill engineering packages reusable procedures; harness engineering wires tools, permissions, and control flow; and loop engineering governs multi-turn evaluation and repair. All five scopes operate on a single medium - the flow of language

5. Semantic geometry and typology-aware evaluation

The preceding sections have focused on language as action and coordination. A complementary question concerns representation and its profound implications on the *explainability of AI*: what is the structure of the meaning space over which prompts, instructions, and agentic procedures operate? Distributional semantics has long argued that meaning emerges from patterns of use (Firth 1957; Boleda 2020; Lenci and Sahlgren 2023). Neural language models extend this idea by embedding linguistic units into high-dimensional spaces, where semantic relations are partially reflected in distances, directions, neighborhoods, and trajectories (Reif et al. 2019).

However, meaning is not only similarity. Human lexical and conceptual systems also contain hierarchy, inclusion, asymmetry, metaphorical projection, cultural categorization, and typological variation. For example, purely flat representations may

be insufficient. Work on hyperbolic and Poincare embeddings has shown that negatively curved spaces can better represent hierarchical semantic structures such as WordNet-like taxonomies (Nickel and Kiela 2017; Ku et al. 2026). Building on this line of work, our own research models culturally embedded semantic fields as structured manifolds rather than flat vector spaces, taking the diachronic development of Chinese *qíng* 情 (“affect, emotion, situation”) as a case study, showing that lexicalization and conceptual extension trace measurable trajectories through a curved semantic space whose local geometry encodes these shifts (Hsieh, in preparation; cf. Figure 3). Such results suggest that semantic geometry is not arbitrary: different linguistic relations may require different geometric assumptions.

Beyond the static geometry of embedding spaces, the depth at which meaning is resolved is itself linguistically structured. Analyzing nonliteral language across all layers of a large model with sparse autoencoders, Yeh et al. (2026) report a lexical familiarity gradient: entrenched multi-word units (idioms) diverge immediately at the first layer, expressions built from individually familiar words (metaphor and semantic-shift slang) are reanalyzed in a narrow early-layer band, and novel forms (neologisms) require deep-layer construction and roughly three times as many representational features. Processing depth thus tracks degrees of entrenchment rather than figurative category - a mechanistic counterpart to usage-based accounts in which lexicalization is a matter of degree. The same study cautions that raw sparse-autoencoder feature counts are confounded by layer-wise activation density, a reminder that interpretability metrics, like linguistic evaluation more generally, require properly matched baselines.



Figure 3. A manifold-style representation of qing as a culturally embedded semantic field

This semantic-geometry point is relevant to agentic AI because prompts and instructions operate over geometric representational space. A directive such as “summarize cautiously,” a vague expression such as “a little closer,” or a governance instruction such as “ask before performing irreversible actions” does not merely add textual content. It reshapes the region of possible interpretations and actions. Pragmatics, in this sense, can be seen as dynamic movement over semantic geometry.

In terms of explainable evaluation, I argue that linguistic typology provides a principled way to probe whether such geometric representations are linguistically adequate. Unlike generic benchmark tasks, typological evaluation can systematically vary phonological, morphological, syntactic, semantic, and discourse properties across languages. It asks not only whether a model can produce the right answer, but whether its internal structure can support generalization across unfamiliar linguistic systems.

This is the motivation behind linguistically informed benchmarks such as

LOBSTER (Lian et al. 2025), which evaluate structured reasoning over problems inspired by the International Linguistics Olympiad. Such tasks are valuable because they cannot be solved by surface pattern matching alone. They require the induction of latent rules, the comparison of unfamiliar forms, the construction of intermediate representations, and the evaluation of reasoning quality. A final answer may be correct by accident; a rule-based explanation reveals more about the model's linguistic competence.

From the present perspective, reasoning can be interpreted as a trajectory through semantic and structural space. Good reasoning follows a coherent path through relevant constraints; poor reasoning jumps across incompatible regions, ignores necessary distinctions, or collapses multiple possible structures into a convenient but false solution. Iterative reasoning and multi-agent critique can improve performance because they explore, refine, and verify multiple trajectories (Yao et al. 2023). Typological diversity is therefore not merely an issue of fairness or coverage, although it is certainly both. It is also a scientific instrument. It allows researchers to test where model geometry is dense, where it is distorted, where it transfers across languages, and where it fails. This matters especially for low-resource languages, non-alphabetic scripts, rich morphology, discourse particles, honorific systems, evidentiality (Aikhenvald 2004), and culturally embedded semantic categories.

The broader implication is that evaluation should move from accuracy alone toward structure-sensitive diagnostics. Such diagnostics may include typological balance, rule-quality assessment, representational neighborhood analysis, curvature-sensitive measures, trajectory evaluation, and discourse-level robustness. These measures are not replacements for task performance, but they provide a deeper account of why a system succeeds or fails.

In this way, typology and semantic geometry form a bridge between linguistics and AI evaluation. Typology supplies the empirical diversity of human language; geometry supplies a formal vocabulary for representational structure; agentic orchestration supplies the action-oriented setting in which language is used to coordinate behavior. The three perspectives together make it possible to study language models not merely as text predictors, but as dynamic systems of meaning, action, and coordination.

6. Implications for linguistics and AI

The framework proposed here has several implications. First, it suggests that pragmatics should move closer to the center of AI research. Many critical failures in agent systems are not failures of grammaticality or lexical meaning. They are failures of deixis, presupposition, implicature, repair, authorization, commitment, and genre expectation. These are precisely the domains in which linguistic pragmatics and discourse analysis have long-standing expertise.

Second, it suggests that language resources should be redesigned for agentic environments. Lexicons, ontologies, WordNets, corpora, typological databases, and discourse annotations can all become more than background data. They can serve as tools for shaping semantic geometry, grounding multilingual evaluation, defining skills, supporting repair, and making agent behavior more inspectable.

Third, it highlights the emergence of new genres in human-agent and agent-agent communication. Prompts, system instructions, skill files, memory notes, tool schemas, agent handoffs, safety policies, and model-generated critiques are not traditional genres, but they are now consequential communicative forms. They deserve linguistic description. Their conventions, failure modes, register variation, and institutional consequences should become objects of study.

Fourth, it reframes AI safety and governance as partly linguistic problems. Safer systems require not only better models, but also clearer instructions, better repair protocols, explicit permission structures, auditable memory, robust handoff conventions, and evaluation methods sensitive to pragmatic transformation. Robustness to non-standard language is one concrete instance: mechanistic analysis shows that the depth at which a model resolves nonliteral expressions is governed by lexical familiarity rather than figurative type, so novel vocabulary demands qualitatively deeper processing and substantially more representational resources than entrenched forms (Yeh et al. 2026). In this sense, language is not merely a surface channel through which AI systems communicate. It is one of the principal media through which they are governed.

7. Conclusion

This article has argued that agentic AI requires an expanded linguistic framework. Language in contemporary AI systems is not only a symbolic system for representing meaning. It is also an interface for action, a medium for memory, a structure in representational geometry, and a coordination protocol for distributed intelligence.

This framework brings together several strands of linguistic theory. Speech act theory helps us understand prompting and tool use as actions. Pragmatics explains why ambiguity, presupposition, deixis, repair, and felicity conditions are central to reliable agent behavior. Distributional and geometric semantics help us interpret model-internal meaning spaces. Typology provides a principled way to evaluate whether those spaces capture the diversity and structure of human languages. Discourse analysis and genre theory help us study new communicative forms emerging in human-agent and agent-agent interaction.

The practical conclusion is equally important. As AI systems become more agentic, the human role will increasingly involve defining problems, specifying norms, designing workflows, writing instructions, managing memory, reviewing outputs, and constructing environments in which agents can coordinate safely. These tasks are all deeply linguistic. Linguistics, therefore, should not be treated as a peripheral humanities discipline in the age of AI. It should be recognized as a central language information science for understanding, designing, and evaluating intelligent systems.

References

- Aikhenvald, Alexandra Y. 2004. *Evidentiality*. Oxford: Oxford University Press.
- Austin, John Langshaw. 1962. *How to do things with words*. Oxford: Clarendon Press.
- Bender, Emily M. and Alexander Koller. 2020. Climbing towards NLU: On meaning, form, and understanding in the age of data. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault (eds.), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 5185–5198. Online. Association for Computational Linguistics. <https://doi.org/10.18653/v1/2020.acl-main.463>
- Boleda, Gemma. 2020. Distributional semantics and linguistic theory. *Annual Review of Linguistics* 6: 213–234. <https://doi.org/10.1146/annurev-linguistics-011619-030303>
- Brown, Tom B., Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal,

- Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (eds.), *Advances in Neural Information Processing Systems* 33, 1877–1901. Red Hook, NY: Curran Associates, Inc.
- Clark, Herbert H. 1996. *Using language*. Cambridge: Cambridge University Press.
- Firth, John R. 1957. A synopsis of linguistic theory, 1930–1955. In John R. Firth (ed.), *Studies in linguistic analysis*, 1–32. Oxford: Basil Blackwell.
- Grice, H. Paul. 1975. Logic and conversation. In Peter Cole and Jerry L. Morgan (eds.), *Syntax and semantics 3: Speech acts*, 41–58. New York, NY: Academic Press.
- Hsieh, Shu-Kai. 2026. Semantic manifolds and the diachronic geometry of emotion words in Chinese. Manuscript. National Taiwan University.
- Koerner, Ernst Frideryk Konrad. 1973. *Ferdinand de Saussure: Origin and development of his linguistic thought in Western studies of language: A contribution to the history and theory of linguistics*. Braunschweig: Vieweg.
- Ku, Mao-Chang, Da-Chen Lian, Pin-Er Chen, Po-Ya Angela Wang, Wei-Ling Chen, and Shu-Kai Hsieh. 2026. When structure matters: Cross-lingual hyperbolic embeddings for Chinese and English Wordnets. In Stelios Piperidis, Núria Bel, Henk van den Heuvel, Nancy Ide, Simon Krek, and Antonio Toral (eds.), *Proceedings of the Fifteenth Language Resources and Evaluation Conference*, 12054–12071. Palma de Mallorca: ELRA Language Resource Association. <https://doi.org/10.63317/55a4sr9mfucq>
- Lenci, Alessandro and Magnus Sahlgren. 2023. *Distributional semantics*. Cambridge: Cambridge University Press.
- Levinson, Stephen C. 1983. *Pragmatics*. Cambridge: Cambridge University Press.
- Lian, Da-Chen, Ri-Sheng Huang, Pin-Er Chen, Chunki Lim, You-Kuan Lin, Guan-Yu Tseng, Zhen-Yu Lin, Pin-Cheng Chen, and Shu-Kai Hsieh. 2025. LOBSTER: Linguistics Olympiad benchmark for structured evaluation on reasoning. In Kai-Wei Chang, Ke-Han Lu, Chih-Kai Yang, Zhi-Rui Tam, Wen-Yu Chang, and Chung-Che Wang (eds.), *Proceedings of the 37th Conference on Computational Linguistics and Speech Processing (ROCLING 2025)*, 193–229. Taipei City: Association for Computational Linguistics.
- Mikolov, Tomas, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.
- Nickel, Maximilian and Douwe Kiela. 2017. Poincaré embeddings for learning hierarchical representations. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems* 30, 6338–6347. Red Hook, NY: Curran Associates, Inc.

- Petrov, Aleksandar, Emanuele La Malfa, Philip H. S. Torr, and Adel Bibi. 2023. Language model tokenizers introduce unfairness between languages. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems* 36, 36963–36990. Red Hook, NY: Curran Associates, Inc. <https://doi.org/10.52202/075280-1608>
- Piantadosi, Steven T. 2024. Modern language models refute Chomsky’s approach to language. In Edward Gibson and Moshe Poliak (eds.), *From fieldwork to linguistic theory: A tribute to Dan Everett*, 353–414. Berlin: Language Science Press. <https://doi.org/10.5281/zenodo.12665933>
- Qu, Changle, Sunhao Dai, Xiaochi Wei, Hengyi Cai, Shuaiqiang Wang, Dawei Yin, Jun Xu, and Ji-Rong Wen. 2025. Tool learning with large language models: A survey. *Frontiers of Computer Science* 19(8): 198343. <https://doi.org/10.1007/s11704-024-40678-2>
- Reif, Emily, Ann Yuan, Martin Wattenberg, Fernanda B. Viégas, Andy Coenen, Adam Pearce, and Been Kim. 2019. Visualizing and measuring the geometry of BERT. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d’Alché-Buc, Emily B. Fox, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems* 32, 8592–8600. Red Hook, NY: Curran Associates, Inc.
- Ruan, Jianhao, Zhihao Xu, Yiran Peng, Fashen Ren, Zhaoyang Yu, Xinbing Liang, Jinyu Xiang, Bang Liu, Chenglin Wu, Yuyu Luo, and Jiayi Zhang. 2026. AOrchestra: Automating sub-agent creation for agentic orchestration. *arXiv preprint arXiv:2602.03786*.
- Sacks, Harvey, Emanuel A. Schegloff, and Gail Jefferson. 1974. A simplest systematics for the organization of turn-taking for conversation. *Language* 50(4): 696–735. <https://doi.org/10.2307/412243>
- Saussure, Ferdinand de. 1916. *Cours de linguistique générale*. Paris: Payot.
- Schegloff, Emanuel A., Gail Jefferson, and Harvey Sacks. 1977. The preference for self-correction in the organization of repair in conversation. *Language* 53(2): 361–382. <https://doi.org/10.2307/413107>
- Schick, Timo, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: Language models can teach themselves to use tools. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems* 36, 68539–68551. Red Hook, NY: Curran Associates, Inc. <https://doi.org/10.52202/075280-2997>
- Schmidt, Craig W., Varshini Reddy, Haoran Zhang, Alec Alameddine, Omri Uzan, Yuval Pinter, and Chris Tanner. 2024. Tokenization is more than compression. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 678–702. Miami, FL: Association for Computational Linguistics. <https://doi.org/10.18653/v1/2024.emnlp-main.40>
- Searle, John R. 1969. *Speech acts: An essay in the philosophy of language*. Cambridge: Cambridge

University Press.

- Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems* 30, 5998–6008. Red Hook, NY: Curran Associates, Inc.
- Yao, Shunyu, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. ReAct: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations (ICLR 2023)*. Kigali: OpenReview.net.
- Yeh, Lang-Ching, Yu-Chieh Wang, and Shu-Kai Hsieh. 2026. Lexical familiarity predicts processing depth for nonliteral language in large language models. In Kai-Wei Chang, Ninareh Mehrabi, Satyapriya Krishna, Anubrata Das, Jwala Dhamala, Yang Trista Cao, Tharindu Kumarage, Anil Ramakrishna, Christos Christodoulopoulos, Yixin Wan, Aram Galystan, Anoop Kumar, and Rahul Gupta (eds.), *Proceedings of the 6th Workshop on Trustworthy NLP (TrustNLP 2026)*, 456–470. San Diego, CA: Association for Computational Linguistics. <https://doi.org/10.18653/v1/2026.trustnlp-main.32>
- Zhang, Charles, Benji Peng, Xintian Sun, Qian Niu, Junyu Liu, Keyu Chen, Ming Li, Pohsun Feng, Ziqian Bi, Ming Liu, Yichao Zhang, Cheng Fei, Caitlyn Heqi Yin, Lawrence KQ Yan, and Tianyang Wang. 2024. From word vectors to multimodal embeddings: Techniques, applications, and future directions for large language models. *arXiv preprint arXiv:2411.05036*.

Shu-Kai Hsieh

Professor

Graduate Institute of Linguistics / Brain and Mind

National Taiwan University

No. 1, Sec. 4, Roosevelt Road,

Taipei, 10617, Taiwan

E-mail: shukaihsieh@ntu.edu.tw

Received: 2026. 06. 21.

Revised: 2026. 08. 03.

Accepted: 2026. 08. 10.