MIT Press • Open Encyclopedia of Cognitive Science

Natural Language Processing

Julia Hockenmaier[1]

[1]University of Illinois at Urbana-Champaign

MIT Press

Published on: Jul 16, 2026

URL: https://oecs.mit.edu/pub/vfg47thc

Natural language processing (NLP) aims to develop systems that can analyze, summarize, translate, or generate text in natural language. NLP applications such as sentiment analysis, machine translation, chatbots, and digital assistants have become widely available in recent years. But how do you actually get a computer to understand or produce natural language? It turns out that much of the progress in NLP, including the recent development of large language models, which have fundamentally changed the field, would be impossible without contemporary hardware and vast volumes of digital text. However, the idea that language understanding or translation could be performed by a machine long predates the invention of electronic computers and had an important influence on philosophy, linguistics, and cognitive science.

History

Precursors and early NLP

Computational approaches to language are at least as old as computers and computer science. The first actual NLP systems, a syntactic parser (Joshi & Hopely, 1996) and the IBM–Georgetown machine translation experiment, were developed in the mid-1950s. Other early NLP systems include ELIZA (Weizenbaum, 1966), the first chatbot; SHRDLU (Winograd, 1970), a natural language interface to a (simulated) robot operating in a blocks world; and Lunar (Woods et al., 1972), a question-answering system that provided access to a database of geological facts.

The notion that language can be described by a finite set of rules was probably first developed by the ancient Sanskrit grammarian Panini. In a similar vein, linguists in the mid-20th century began to develop a variety of more formal theories of grammar (Ajdukiewicz, 1935; Chomsky, 1957; Tesniere, 1959). The related question of how expressive (grammar) formalisms have to be in order to be sufficient for natural language had a significant impact on the development of formal language theory and remained an important topic of study in NLP and computational linguistics for a long time.

The statistical and distributional properties of language were studied both by mid-20th century linguists (Firth, 1957; Harris, 1960; Zipf, 1935) and by WWII codebreakers and telecommunication experts such as Claude Shannon and Alan Turing. Shannon (1948) first defined a *language model* as a probability distribution over the possible strings (sentences) of a language and introduced the so-called *noisy channel model*, both of which formed the basis of many approaches to statistical NLP in the 1990s (see below). Turing (1950) proposed to use natural language dialogue as a test of machine intelligence [see The Turing Test].

Symbolic NLP

Until the statistical revolution in the 1990s, NLP systems relied on various kinds of purely symbolic representations. Representations to capture the *structure* of words and sentences were often based on various types of formal automata (some examples of these include finite-state approaches for morphology and phonology or augmented transition networks for syntax; Kaplan & Kay, 1994;

Koskenniemi, 1983; Woods, 1970), or unification-based feature structures, and included the development of a wide variety of different grammar formalisms (Gazdar, 1985; Joshi, 1988; Kaplan & Bresnan, 1982; Pollard & Sag, 1994; Steedman, 2000), several of which were later employed in large-scale grammar engineering efforts. The structure of discourse and dialogue also received significant attention (Grosz et al., 1995; Grosz & Sidner, 1986; Mann & Thompson, 1988).

At the same time, a variety of representations to capture the linguistic and common sense knowledge required to understand the *meaning* of words and sentences were proposed, including case grammar (Fillmore, 1968), scripts (Schank & Abelson, 1977), semantic networks, and specialized knowledge representation languages. These approaches later led to the development of semantically annotated corpora such as FrameNet (Baker et al., 1998) and the Proposition Bank (Palmer et al., 2005) as well as large-scale lexical ontologies such as WordNet (Fellbaum, 1998) and knowledge graphs such as DBpedia or Freebase. Semantic networks, ontologies, and knowledge graphs capture semantic relations such as hyponymy ("is-a") or meronymy ("has-a," parts-of) between words, concepts, or real-world objects. NLP continued to shape and be influenced by developments in other parts of artificial intelligence, for example, logic programming and planning (which became important for natural language generation; see, e.g., Reiter & Dale, 2000).

Statistical NLP

Systems that rely purely on manually developed rules are not just difficult to maintain and extend but also have no principled mechanism to handle ambiguity.

The practical advantages of probabilistic approaches were first shown by Fred Jelinek and Bob Mercer's team at IBM for speech recognition and machine translation (Brown et al., 1993). One reason why machine translation was among the first areas of NLP to turn to statistical models is that it could leverage pre-existing parallel corpora of text translated into one or more additional languages (e.g., parliamentary debates in localities with multiple official languages such as Canada or Hong Kong). The broader paradigm shift towards statistical NLP was also enabled by the availability of *linguistically annotated corpora* such as the part-of-speech (POS)-tagged Brown Corpus (Kučera et al., 1967), the Penn Treebank (Marcus et al., 1993), and the Prague Dependency Treebank (Hajič et al., 2020) for syntactic parsing and the Proposition Bank (Palmer et al., 2005) for semantic role labeling. It soon became clear that many NLP tasks can be framed as *classification*, *sequence labeling*, or *structure prediction* problems: given an input text, predict a single label (e.g., a sentiment label), one label for each word (e.g., POS tags), or a complex object (e.g., a syntactic parse tree). Because annotated corpora provided the desired labels or structures for relatively large amounts of data, much of NLP became amenable to supervised machine learning and statistical modeling (Manning & Schütze, 1999).

This shift towards NLP as an empirical discipline was also driven by a series of government-sponsored programs in the United States (the Message Understanding Conferences and the Defense Advanced Research Projects Agency's TIPSTER program) as well as so-called "shared tasks" associated with

workshops such as Computational Natural Language Learning and Senseval/SemEval. That is, NLP became an experimental field in which progress on a problem (typically framed as a "task") is measured by comparing the accuracy of different models that are trained and tested under the same conditions. Each such evaluation or task requires a sufficiently large dataset (corpus) that is (accurately) annotated with the linguistic information to be predicted, split into disjoint test and training sets, and given an appropriate set of evaluation metrics to measure how well the annotations predicted by models on the test data match the gold standard human annotations. This focus on empirical evaluation and on reproducibility has, on the whole, been instrumental in fostering progress in NLP.

The de facto standardization of problem definitions, annotation schemes, and evaluation metrics made it possible to directly compare different approaches, whereas training and evaluating models on large amounts of real-world data led to significantly increased coverage and robustness. However, this paradigm shift has also resulted in so-called "state-of-the-art chasing," that is, the notion that any improvement in some test score corresponds to a (publishable) contribution and that approaches that do not yield such an improvement are not worthy of consideration. Linguistically interesting, but rare, corner cases that could inform linguistic theory are often neglected in this process.

Neural NLP and large language models

Breakthroughs in computer vision in the mid-2010s led to a resurgence of interest in neural networks, now under the moniker of "deep" learning (because of the depth, or large number of layers, in state-of-the-art networks). Visual data (images) and visual processing are quite naturally formalized in terms of the real-valued matrices and matrix operations that form the basis of neural networks. However, neural networks seemed less applicable to modeling natural language, that is, sequences of words drawn from a very large, discrete vocabulary. This changed when it was shown that neural networks could learn to map words to vectors in such a way that some semantic relations between words were captured (Mikolov et al., 2013) and that recurrent networks and transformers that use such vectors as input would generally outperform state-of-the-art statistical models on many standard NLP tasks (Devlin et al., 2019; Peters et al., 2018) [see Recurrent Neural Networks; Transformers]. These end-to-end neural models were typically pretrained on large amounts of raw text and then fine-tuned on standard benchmarks for each NLP task. This was followed by demonstrations that models that were pretrained on even larger amounts of text, often followed by further training based on human feedback (Ouyang et al., 2022), could lead to even better performance on a wide variety of tasks (Brown et al., 2020). In fact, training at scale seemed to obviate the need to fine-tune (adapt) these models for each task, instead relying on prompting these models with a few examples (under the misleadingly named "in-context learning" framework). This has set off a race for industry labs to produce ever bigger *large language models (LLMs)* and publicly available chatbots that can not just generate text in a vast variety of genres and languages, translate between natural languages, or summarize entire articles but also read and generate code [see Large Language Models].

Core concepts

Classical NLP tasks and the NLP pipeline

Before so-called end-to-end neural models became prevalent in the 2020s, many NLP systems would contain separate components that each perform a particular task. Because these components typically process the input in a particular sequence, this is typically referred to as the classical *NLP pipeline*. During this process, the input text is augmented with more and more additional annotations that represent different types of linguistic information.

First, a *tokenizer* splits input text into individual sentences and words (tokens) to be processed. A *POS tagger* then labels the tokens in a sentence with labels (drawn from a fixed POS tag set) that indicate whether they serve as a noun, verb, adjective, or another part of speech in this sentence. A *named entity (NE) tagger* operates much like a POS tagger but identifies sequences of tokens that correspond to the names of entities of interest (e.g., people, locations, organizations, and dates for the news domain or genes, proteins, and diseases for biomedical papers). A *syntactic parser* reads in a sentence (typically POS tagged) and identifies its most likely grammatical structure (parse), whereas a *semantic parser* translates sentences (that may be syntactically parsed) into a meaning representation.

For applications in which the input consists of multiple sentences (e.g., an entire newspaper article or other document), further processing may be necessary. A *coreference resolution* component reads in the entire input text (which is likely already POS tagged, NE tagged, and possibly parsed) and identifies which noun phrases refer to the same entities. In an *information extraction* system, a *relation extraction* component may then read in the entire input text to identify particular relations of interest between the entities mentioned in the text, whereas an *event extraction* component may identify mentions of particular events of interest (possibly involving one or more entities). In information extraction, document understanding has often been framed as the task of filling in predefined *templates* that contain slots about the type of event, involved entities, times, dates, and locations mentioned in a document.

Dialog systems need to process individual user utterances, which are much shorter and less well formed than news articles or other written documents, including interruptions, interjections (“hmm,” “uhm”), speech repairs (“today—I mean tomorrow”), and other disfluencies. In traditional *task-based dialog systems* (used, e.g., to book air travel or, more generally, for customer support), the natural language understanding task typically consists of identifying *user intent* (what task does the user want to accomplish?), the particular *dialog act* (is the user asking a question, providing new information, or confirming a system choice?), and, because each task (e.g., booking a flight) is typically associated with a particular template (*frame*), which specific *slots* can be filled (e.g., day of departure, airline, etc.).

Each of the abovementioned components assumes its input and/or output is based on a specific symbolic representation (e.g., a particular inventory of POS tags, NE tags, dialog acts, frames, and slots) and typically relies on a model that has been trained on a corpus that is annotated with this representation.

One challenge with this pipeline-based approach is that errors propagate, and mistakes compound. This makes such systems inherently brittle.

By the late 2010s, so-called end-to-end neural models (based on recurrent neural networks or transformers) could directly perform any of the core NLP tasks on raw text and outperformed systems based on the traditional pipeline on standard NLP benchmarks.

Core concepts in statistical NLP

Formally, a language can be seen as a (typically infinite) set of sequences of words drawn from a (finite or infinite) vocabulary, and a grammar may define what sequences of words from this vocabulary belong to the language ("John eats a pizza" is an English sentence, "Pizza John eats a" is not). A *language model* assigns probabilities to these word sequences, typically by predicting the probability of the next word given the start of the sentence—$P(w_1 = \textit{"John"})$—or an initial sequence of words, for example, $P(w_2 = \textit{"eats"} \,|w_1 = \textit{"John"})$, $P(w_3 = \textit{"a"} \,|w_1 w_2 = \textit{"John eats"})$, etc., and multiplying these probabilities together to yield the probability of the entire sequence. When these probabilities are based on word co-occurrence counts observed in real text, such models can be used to distinguish grammatical from ungrammatical and more fluent from less fluent text. However, the number of possible word sequences above a certain length becomes astronomical, and even with very large amounts of text, we cannot reliably estimate the probability of the words that may follow each of them. Instead, we use so-called *n*-gram models, in which the probability of each word is only conditioned on the preceding *n*-1 words (with *n* ranging from three to six or seven).

Statistical models can also be used as a principled mechanism to handle ambiguity and uncertainty. For example, each French sentence can have many possible English translations. We can therefore formalize French-to-English machine translation as the task of finding the best, or most likely, English translation E^* of the French input F : $E^* = argmax_E \, P(E|F)$. Using Bayes' rule, this task is equivalent to finding an English sentence that is both highly fluent and whose most likely translation into French would be $F : E^* = argmax_E \, P(F|E)P(E)$ [see Bayesianism]. The advantage of this reformulation is twofold: first, we can leverage English language models $P(E)$ that are trained on large amounts of English text to make sure *E* is fluent, and second, because *F* is already known, we can use a much simpler translation model for $P(F|E)$ because we only need to capture that *F* is a faithful translation of *E* (e.g., by making sure the words or phrases in *F* are aligned to words or phrases in *E*). This is also an application of the noisy channel model mentioned above; according to this metaphor, the French speaker encoded the original message *E* into French as *F*, and the translation task is to decode *E* from *F*.

Similar arguments can be used to motivate, for example, framing syntactic parsing as the task of finding the best parse tree *T* for a sentence *S* by using a probabilistic grammar that assigns probabilities to all parse trees in the language and then returning the tree with the highest probability among all the possible trees for *S*.

Core concepts in neural NLP

How do you process language with neural networks, a framework that assumes inputs and outputs consist of arrays of real numbers (vectors, matrices, and tensors) [see Distributional Semantics]? Language consists of arbitrarily long sequences of words drawn from a very large, discrete vocabulary V. Assuming the vocabulary is finite, one way to represent words as vectors is with a naive "one-hot" representation, in which each the i-th word in the vocabulary ("lexical item") is represented as a $|V|$-dimensional vector in which the i-th element is "turned on" (equal to 1), and all other elements are off (set to 0), so that the first word (say, the article "a") in a 10,000-word dictionary is represented as a 1 followed by 9,999 zeros ([1, 0,...,0]), and the last word in the dictionary (say, "zzz") is represented as 9,999 zeros followed by a 1: [0,...,0, 1]. Other ways to represent words as vectors arose from early work in information retrieval and in lexical semantics, in which so-called *distributional similarities* between words are based on the idea that words that appear in similar contexts have similar meanings, and words are represented as vectors whose elements indicate how likely they are to appear in a particular context (typically defined as being in the vicinity of a specific word). Because of the large number of possible contexts, this approach yields sparse, high-dimensional vectors. This changed with the development of embedding models such as word2vec (Mikolov et al., 2013) and GloVe (Pennington et al., 2014), which use neural networks to map each word in the vocabulary to a dense, relatively low-dimensional (typically, $n = 300$) vector.

Processing sentences or documents (i.e., arbitrarily long sequences of words) with a neural network requires a shift away from basic feedforward neural networks (which require fixed-size inputs) to either convolutional, recurrent, or transformer-based architectures. Convolutional networks pass (fixed-size) sliding windows of k tokens over the input, such that deeper layers see increasingly longer token sequences at a time, eventually culminating in a single fixed-size representation of the entire input. This makes them well suited for text classification tasks.

Recurrent architectures and transformers read input or generate output one token at a time. They can either return a label for each input token, making them suitable for sequence labeling tasks such as part-of-speech tagging. They can also be used as language models and predict the probability of the next token or return the most likely next token at each time step. This capability to generate arbitrarily long output texts conditioned on arbitrarily long inputs makes both recurrent networks and transformers well suited for so-called sequence-to-sequence tasks such as machine translation, summarization, or answer generation in dialogue.

Recurrent networks update a single fixed-size memory state of the entire sequence at each time step, whereas transformers use attention mechanisms that allow them to consider all tokens seen or generated so far. This makes transformers generally more robust than recurrent architectures, especially for longer input or output sequences. Attention mechanisms are inspired by the notion of word alignment in machine translation and represent the entire input seen so far as a weighted average of the previous tokens (allowing the weights to change based on the current input so that the model "pays more attention" to currently relevant prior input tokens).

Why is NLP challenging?

Healthy adults understand and speak their native language effortlessly, even though any language's vocabulary is vast and constantly changing, and words from that vocabulary can be combined in unbounded ways to form arbitrarily long and complex utterances. However, systems that accurately understand natural language (i.e., that are able to correctly infer the meaning and structure of any such utterance) and that can produce (generate) fluent, contextually appropriate language need to maintain the knowledge required to do so in finite memory. This observation underlies the principle of compositionality and was a core motivation for phrase structure grammars [see Compositionality]. Adequately addressing it is both helped and hindered by the long-tailed (Zipfian) nature of many linguistic phenomena, including the distribution of words. Additionally, systems that understand or produce natural language need to be able to handle input that is ambiguous or vague and to infer information that is implied, but not explicitly stated, in the input.

Core challenge one: Achieving coverage and robustness in the face of the long tail

The distribution of words (and likely that of many other linguistic phenomena, such as different types of syntactic constructions) follows a power law distribution (Zipf's law), with a relatively small number of highly frequent words (mostly function words such as pronouns, prepositions, determiners, or conjunctions) and a potentially unbounded number of increasingly rare words (mostly content words such as nouns, verbs, adjectives, and adverbs as well as numbers and, depending on the language, a myriad of inflected or compounded word forms). On the one hand, this has an obvious advantage both for NLP and for human language learners: a small, core vocabulary can often be sufficient to roughly understand much of a text. On the other hand, the long tail of rare words poses fundamental challenges because a precise understanding will typically require knowledge of some word forms that are either so rare or so novel that the system or learner has either never seen them before or has perhaps encountered them only in a context in which they had a different meaning or (in a lightly inflected language such as English) part of speech.

Previously unseen word forms may arise in all genres, both in spoken or informal language (including more recent genres such as text messages or social media posts, which have their own, rapidly evolving conventions) and also in well-edited texts, such as literature, news, or scientific articles [see Linguistic Variation]. However, infinite vocabularies cannot be stored in finite memory, and to process unseen text, any probabilistic language model needs to be able to assign nonzero probabilities to words it has not encountered during training. One standard solution for this problem is to replace any rare words during language model training (and any unknown words when using the trained model) with a special *unknown word token* (customarily referred to as "UNK"), whereas so-called *subword tokenization* (which splits longer, infrequent words into common character sequences that do not necessarily correspond to linguistically correct morphemes) has enabled more recent neural models to handle larger vocabularies reliably and robustly.

Core challenge two: Dealing with ambiguity and vagueness

Much work in data-driven, empirical NLP has been driven by the motivation to handle ambiguity in a principled manner and to increase the coverage of systems (i.e., to make them more robust). *Ambiguity* is pervasive in natural language and arises at all levels. Words can have multiple parts of speech (e.g., "place" as a verb or noun) and meanings (e.g., "bank" as a company, a building, a repository, or the side of a river). Sentences may have an exponential number of possible syntactic structures (the textbook example for English is prepositional phrase attachment, i.e., what a preposition modifies, as in, "I see the man with the telescope on the hill"). Across sentences, pronouns, or other referring expressions can have multiple possible antecedents, as exemplified by the so-called Winograd schema in which the most likely antecedent depends on other information (e.g., "The city councilmen refused the demonstrators a permit because they [feared/advocated] violence"; Winograd, 1972). Ambiguity is particularly challenging for NLP systems that use different symbolic representations for each possible interpretation because there is typically only one representation that corresponds to the intended interpretation (which is what distinguishes ambiguity from *vagueness*, which is involved in, e.g., the interpretation of gradable adjectives like "expensive" or "close"; see, e.g., Kennedy, 2011).

Core challenge three: Identifying implicit information

Another core challenge for NLP is the fact that much of the information conveyed in an utterance may not be explicitly stated because it is assumed that the reader or listener will automatically draw the corresponding inferences. *Lexical entailments* are implied by the definition of individual words. They include hyponym–hypernym relations (a "poodle" is a kind of "dog"), which are captured in ontological resources such as WordNet. More general forms of entailment (e.g., that you get wet when you walk in the rain) require *common sense knowledge*. Finally, *implicatures* are inferences that may depend on the (social) context in which something was said (or perhaps omitted), for example, a recommendation letter that only praises a candidate's punctuality.

Questions, controversies, and new developments

Core questions for symbolic NLP: Which representation schemes to use?

The question of what information a symbolic representation should capture and how best to do so (e.g., in a way that is compact or allows for better abstractions or generalizations) gained additional importance as statistical NLP relied increasingly on linguistically annotated corpora for supervised learning. There are typically many different ways to represent linguistic information, varying in the amount of detail captured and the particular analyses adopted. For example, the structure of a sentence such as, "I eat sushi with chopsticks," can be expressed as a phrase structure tree by bracketing constituents such as noun phrases ("I," "sushi," or "chopsticks"), verb phrases ("eat sushi with chopsticks") or prepositional phrases ("with chopsticks"). Alternatively, the structure can be indicated by word–word dependencies that capture grammatical roles ("I" and "sushi" are the subject and direct object of "eat").

The empirical utility of a particular annotation standard (consisting both of the choice of formalism, e.g., dependency or phrase structure grammar, and detailed annotation guidelines, specifying the inventory and usage of dependency or constituent labels) can only be assessed once a sufficiently large amount of data has been annotated so that models can be trained, evaluated, and integrated into larger systems. However, linguistic annotation is labor intensive and requires a significant amount of specialized expertise, which makes it expensive (and often difficult to fund). Therefore, many tasks became defined by a few datasets that were large enough to test and train models on; statistical parsing focused initially on Penn Treebank–style phrase structure trees (ignoring the additional annotations that capture nonlocal dependencies that later enabled the conversion into linguistically expressive grammar formalisms) but has in recent years mostly shifted to dependency-based approaches because of the development of treebanks for many languages in the universal dependency framework (de Marneffe et al., 2021) and the ease with which word–word dependencies can be interpreted for further analysis.

The changing role of corpora and benchmarks

Datasets such as the Brown Corpus (Kučera et al., 1967) and other, later corpora were essential for the development of NLP into a largely empirical discipline. With the rise of the web, large amounts of electronic text became easily available. Non-copyrighted (Creative Commons Licensed) resources such as Wikipedia and Wikidata became especially important for many tasks. In addition to collections of texts that were originally written for other purposes, test suites that have been expressly created to cover a variety of linguistic phenomena of interest (Lehmann et al., 1996) provide a complementary tool for evaluation. Crowdsourcing services enable researchers to create their own corpora, often for purposes in which no existing text data is available at scale. This has enabled new tasks and datasets, for example, for image caption generation (Rashtchian et al., 2010; Hodosh et al., 2013), which could then be used in their own right for new benchmarks (Bowman et al., 2015).

Have LLMs "solved" NLP?

The paradigm shifts towards end-to-end neural models and LLMs have undoubtedly led to significant improvements in performance. Moreover, so-called *multimodal models* that capture both text and images or video are becoming increasingly capable and useful, enabling more and more applications of NLP and speech technologies to so-called *embodied tasks*, including robotics or gaming.

Because LLMs are now widely used in practice, new tasks such as preventing them from *hallucinating* (generating factually incorrect statements) or identifying hallucinations or other factually incorrect statements (including so-called fake news) become increasingly important. However, although the largest (also called "frontier") LLMs can handle an astonishingly wide range of tasks, smaller models that can be fine-tuned are often more suitable for specific tasks. There are also indications that LLMs suffer from data contamination (Carlini et al., 2021), leading to inflated performance on publicly available benchmarks. The scale of data these models are trained on raises ethical and legal questions around potential copyright violations as well as more fundamental scientific questions: Is it really necessary to ingest all available

text ever produced by humanity, or can we achieve similar levels of generalization through smaller but smarter models?

Broader connections

NLP, computational linguistics, and linguistics

NLP is an intellectually rich and highly interdisciplinary field. It is primarily an engineering discipline; from its beginning, much work in NLP has been motivated and funded by its promise for practical applications, either for commercial or intelligence purposes. However, its development is also tightly intertwined with, and often indistinguishable from, that of computational linguistics, "The scientific study of language from a computational perspective" (ACL, n.d.). Computational linguistics has developed alongside modern approaches to theoretical/formal linguistics (including morphology, syntax, semantics, pragmatics, and discourse), corpus linguistics, sociolinguistics, and psycholinguistics. However, much work in theoretical linguistics since Noam Chomsky (1957) has been based on introspective grammaticality judgments of individual sentences by native speakers instead of corpus-based analyses, and Chomsky's antipathy toward empirical and statistical approaches is likely a major contributing factor as to why NLP rediscovered statistics only in the 1990s (Pereira, 2000). In addition to linguistics, both computational linguistics and NLP have always drawn insights from, and contributed to, research in logic and formal language theory as well as statistics, information theory, and machine learning, although the relative importance of these fields has varied over time.

NLP and artificial intelligence

Although NLP has long seen itself as somewhat independent from artificial intelligence, with different professional societies and conferences, NLP is also inextricably linked to the development of artificial intelligence, which sees language as a hallmark of intelligence and reasoning.

NLP and speech processing

Most work in NLP (and computational linguistics) focuses on written or transcribed text (the signal processing techniques required to transcribe and generate audible language, or speech, are beyond the scope of this article) [see Speech Recognition].

Computational models of language acquisition and computational psycholinguistics

Although much of NLP is an applied discipline, there have always been significant strands of inquiry into using computational approaches to model human language acquisition from learning the meaning of words to the induction of syntactic or semantic grammars and to model human sentence processing (e.g., to use parsing models to predict reading times) [see Computational Models of Language Learning; Language Acquisition; Psycholinguistics; Sentence Processing; Word Learning].

Further reading

- Eisenstein, J. (2019). *Introduction to natural language processing*. MIT Press.
- Jurafsky, D., & Martin, J. H. (2026). *Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition with language models* (3rd ed.). Stanford University Press

References

- ACL. (n.d.). *What is the ACL and what is computational linguistics?* Association for Computational Linguistics. Retrieved September 9, 2025, from https://www.aclweb.org/portal/what-is-cl

  ↩
- Ajdukiewicz, K. (1935). Die syntaktische Konnexität. In S. McCall (Ed.), *Polish logic 1920-1939* (pp. 207–231). Oxford University Press.

  ↩
- Baker, C. F., Fillmore, C. J., & Lowe, J. B. (1998). The Berkeley FrameNet project. *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics*, *1*, 86–90. https://doi.org/10.3115/980845.980860

  ↩
- Bowman, S. R., Angeli, G., Potts, Christopher, & Manning, C. D. (2015). A large annotated corpus for learning natural language inference. *arXiv*. https://doi.org/10.48550/arXiv.1508.05326

  ↩
- Brown, P. F., Della Pietra, S. A., Della Pietra, V. J., & Mercer, R. L. (1993). The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, *19*(2), 263–311. https://aclanthology.org/J93-2003/

  ↩
- Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., … Amodei, D. (2020). Language models are few-shot learners. *arXiv*. https://doi.org/10.48550/arXiv.2005.14165

  ↩
- Carlini, N., Tramèr, F., Wallace, E., Jagielski, M., Herbert-Voss, A., Lee, K., Roberts, A., Brown, T., Song, D., Erlingsson, Ú., Oprea, A., & Raffel, C. (2021). Extracting training data from large language models. *arXiv*. https://doi.org/10.48550/arXiv.2012.07805

  ↩

- Chomsky, N. (1957). *Syntactic structures*. Mouton.

  ↩

- de Marneffe, M.-C., Manning, C. D., Nivre, J., & Zeman, D. (2021). Universal dependencies. *Computational Linguistics*, *47*(2), 255–308. https://doi.org/10.1162/coli_a_00402

  ↩

- Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In J. Burstein, C. Doran, & T. Solorio (Eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human language technologies* (Vol. 1, pp. 4171–4186). Association for Computational Linguistics. https://doi.org/10.18653/v1/N19-1423

  ↩

- Fellbaum, C. (Ed.). (1998). *WordNet: An electronic lexical database*. MIT Press.

  ↩

- Fillmore, C. J. (1968). The case for case. In E. W. Bach & R. T. Harms (Eds.), *Universals in linguistic theory* (pp. 1–88). Holt, Rinehart, and Winston.

  ↩

- Firth, J. R. (1957). *Studies in linguistic analysis.* Blackwell.

  ↩

- Gazdar, G. (1985). *Generalized phrase structure grammar.* Harvard University Press.

  ↩

- Grosz, B. J., & Sidner, C. L. (1986). Attention, intentions, and the structure of discourse. *Computational Linguistics*, *12*(3), 175–204. https://aclanthology.org/J86-3001/

  ↩

- Grosz, B. J., Joshi, A. K., & Weinstein, S. (1995). Centering: A framework for modeling the local coherence of discourse. *Computational Linguistics*, *21*(2), 203–225. https://aclanthology.org/J95-2003/

  ↩

- Hajič, J., Bejček, E., Hlavacova, J., Mikulová, M., Straka, M., Štěpánek, J., & Štěpánková, B. (2020). Prague Dependency Treebank—Consolidated 1.0. In N. Calzolari, F. Béchet, P. Blache, K. Choukri, C. Cieri, T. Declerck, S. Goggi, H. Isahara, B. Maegaard, J. Mariani, H. Mazo, A. Moreno, J. Odijk, & S. Piperidis (Eds.), *Proceedings of the Twelfth Language Resources and Evaluation Conference* (pp. 5208–5218). European Language Resources Association. https://aclanthology.org/2020.lrec-1.641/

  ↩

- Harris, Z. S. (1960). *Structural linguistics* (1st ed.). University of Chicago Press.

↩

- Hodosh, M., Young, P., & Hockenmaier, J. (2013). Framing image description as a ranking task: Data, models and evaluation metrics. *Journal of Artificial Intelligence Research*, *47*, 853–899. https://doi.org/10.1613/jair.3994

↩

- Joshi, A. K. (1988). Tree adjoining grammars. In D. Dowty, L. Karttunen, & A. Zwicky (Eds.), *Natural language parsing* (pp. 206–250). Cambridge University Press.

↩

- Joshi, A. K., & Hopely, P. (1996). A parser from antiquity. *Natural Language Engineering*, *2*(4), 291–294. https://doi.org/10.1017/S1351324997001538

↩

- Kaplan, R. M., & Bresnan, J. (1982). Lexical-functional grammar: A formal system for grammatical representation. In J. Bresnan (Ed.), *The mental representation of grammatical relations* (pp. 173–281). MIT Press.

↩

- Kaplan, R. M., & Kay, M. (1994). Regular models of phonological rule systems. *Computational Linguistics*, *20*(3), 331–378. https://aclanthology.org/J94-3001/

↩

- Kennedy, C. (2011). Ambiguity and vagueness: An overview. In C. Maienborn, K. von Heusinger, & P. Portner (Eds.), *Semantics: An international handbook of natural language meaning* (Vol. 1, pp. 535–574). De Gruyter Mouton.

↩

- Koskenniemi, K. (1983). *Two-level morphology: A general computational model for word-form recognition and production* (Vol. 11). University of Helsinki, Department of Linguistics.

↩

- Kučera, Henry., Francis, W. N., Twaddell, W. F., Marckworth, M. L., Bell, L. M., & Carroll, J. B. (1967). *Computational analysis of present-day American English.* Brown University Press.

↩

- Lehmann, S., Oepen, S., Regnier-Prost, S., Netter, K., Lux, V., Klein, J., Falkedal, K., Fouvry, F., Estival, D., Dauphin, E., Compagnion, H., Baur, J., Balkan, L., & Arnold, D. (1996). TSNLP - Test suites for natural language processing. *arXiv*. https://doi.org/10.48550/arXiv.cmp-lg/9607018

↩

- Mann, W. C., & Thompson, S. A. (1988). Rhetorical structure theory: Toward a functional theory of text organization. *Text & Talk*, *8*(3), 243–281. https://doi.org/10.1515/text.1.1988.8.3.243

↩

- Manning, C., & Schütze, H. (1999). *Foundations of statistical natural language processing*. MIT Press.

↩

- Marcus, M. P., Santorini, B., & Marcinkiewicz, M. A. (1993). Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, *19*(2), 313–330. https://aclanthology.org/J93-2004/

↩

- Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In C. J. Burges, L. Bottou, M. Welling, Z. Ghahramani, & K. Q. Weinberger (Eds.), *Advances in neural information processing systems* (Vol. 26). Curran Associates.

↩

- Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., Schulman, J., Hilton, J., Kelton, F., Miller, L., Simens, M., Askell, A., Welinder, P., Christiano, P. F., Leike, J., & Lowe, R. (2022). Training language models to follow instructions with human feedback. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, & A. Oh (Eds.), *Advances in neural information processing systems* (Vol. 35, pp. 27730–27744). Curran Associates.

↩

- Palmer, M., Gildea, D., & Kingsbury, P. (2005). The proposition bank: An annotated corpus of semantic roles. *Computational Linguistics*, *31*(1), 71–106. https://doi.org/10.1162/0891201053630264

↩

- Pennington, J., Socher, R., & Manning, C. (2014). GloVe: Global vectors for word representation. In A. Moschitti, B. Pang, & W. Daelemans (Eds.), *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing* (pp. 1532–1543). Association for Computational Linguistics.

↩

- Pereira, F. (2000). Formal grammar and information theory: Together again? *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, *358*(1769), 1239–1253. https://doi.org/10.1098/rsta.2000.0583

↩

- Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). Deep contextualized word representations. In M. Walker, H. Ji, & A. Stent (Eds.), *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human language technologies* (Vol. 1, pp. 2227–2237). Association for Computational Linguistics.

  ↩
- Pollard, C., & Sag, I. A. (1994). *Head-driven phrase structure grammar.* University of Chicago Press.

  ↩
- Rashtchian, C., Young, P., Hodosh, M., & Hockenmaier, J. (2010). Collecting image annotations using Amazon's Mechanical Turk. In C. Callison-Burch & M. Dredze (Eds.), *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk* (pp. 139–147). Association for Computational Linguistics.

  ↩
- Reiter, E., & Dale, R. (2000). *Building natural language generation systems.* Cambridge University Press.

  ↩
- Schank, R. C., & Abelson, R. P. (1977). *Scripts, plans, goals, and understanding: An inquiry into human knowledge structures.* Lawrence Erlbaum Associates.

  ↩
- Shannon, C. E. (1948). *A mathematical theory of communication*. University of Illinois Press.

  ↩
- Steedman, M. (2000). *The syntactic process.* MIT Press.

  ↩
- Tesniere, L. (1959). *Elements de syntaxe structurale*. Klincksieck.

  ↩
- Turing, A. M. (1950). I.—Computing machinery and intelligence. *Mind*, *LIX*(236), 433–460. https://doi.org/10.1093/mind/LIX.236.433

  ↩
- Weizenbaum, J. (1966). ELIZA—A computer program for the study of natural language communication between man and machine. *Communications of the ACM*, *9*(1), 36–45. https://doi.org/10.1145/365153.365168

  ↩

- Winograd, T. (1970). *Procedures as a representation for data in a computer program for understanding natural language* [Doctoral dissertation, Massachusetts Institute of Technology]. DSpace@MIT. https://dspace.mit.edu/handle/1721.1/15546

  ↩

- Winograd, T. (1972). Understanding natural language. *Cognitive Psychology*, *3*(1), 1–191. https://doi.org/https://doi.org/10.1016/0010-0285(72)90002-3

  ↩

- Woods, W. A. (1970). Transition network grammars for natural language analysis. *Communications of the ACM*, *13*(10), 591–606. https://doi.org/10.1145/355598.362773

  ↩

- Woods, W. A., Kaplan, R. M., & Nash-Webber, B.-L. (1972). *The lunar sciences natural language information system: Final report.* Bolt, Beranek and Newman

  ↩

- Zipf, G. K. (1935). *The psycho-biology of language: An introduction to dynamic philology.* Houghton Mifflin.

  ↩