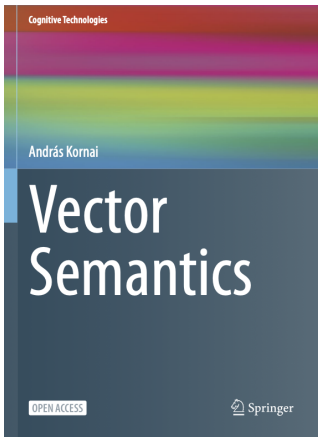
The linguistic pitch

András Kornai
SZTAKI Computer Science Research Institute
and
Dept of Algebra, Budapest University of Technology and Economics

AITP, September 4 2024

Making the pitch for three (interlinked) things

- **Linguistics** This is like making a pitch for physics, so it can be very brief: linguistics is a science, has vast body of empirical data, has considerable body of good and not so good theories, some of which seems pertinent to AI/TP concerns
- **LLMs** The zeitgeist is clear: get with the program! But what program, exactly?
- **Vector semantics**: the book 

and the idea

Why theorem provers should care about natural language

- In mathematics, we have a few axioms and long proofs, this is the natural home of (AI)TP work
- In linguistics, we have many axioms and short proofs, this is the natural home of both syntax and semantics
- Nevertheless, the two setups are basically the same, with highly controlled, mechanistic deduction steps leading from axioms (postulates, lemmas, 'already givens') to novel results (new theorems, sentences)

Syntax

- Generative grammars come in a large variety (about a hundred well worked out formalisms, some very well known, some less so)
- Generally operate on strings, starting with a 'start symbol' (single axiom)
- Also systems of tree rewriting
- Occasionally more complex structures (graphs, hypergraps)
- Very strong connection to automata and semigroups – see Strobl et al., 2024 for a survey of LLMs from the formal language theory perspective
- In the focus of linguistic applications we see transducers (often weighted), in the bulk of day-to-day work LLMs have not replaced finite automata/transducers. `grep` does not hallucinate

Large Language Models

- LLMs do syntax really well, producing sentences that are clearly grammatical
- In fact their performance is superhuman: 100+ languages, with very reasonable translation across them, ability to mimick styles, write poems, etc.
- They are good about assigning grammatical structure (tree diagram, parts-of-speech labels, etc) to sentences
- But not particularly about the classic generative grammar task, which was to decide whether a string is grammatical or not
- Syntax is an AITP subtask (fast provers for grammaticality) but not a very exciting one (most grammar formalisms have a polynomial decision procedure)

Major developments propelling LLMs

- Vectors
  - Key enabler: word vectors (Schütze, 1993) (but goes back to Firth, 1957)
  - First implementation that really worked (Bengio et al., 2003)
  - NLP "almost from scratch" POS, CHUNK, NER, role labeling (Collobert et al., 2011)
  - Has linear structure (king–queen=man–woman) (Mikolov, Yih, and Zweig, 2013)
  - Why? (Pennington, Socher, and Manning, 2014; Arora et al., 2015; Gittens, Achlioptas, and Mahoney, 2017)
- Subword units
- Neural nets
- Attention

Subword units

- The basic units are words, but linguists have been using subword units called *morphemes* for the longest time
- These are the smallest units to which meaning can be attributed
- LLMs actually use meaningless subword units (roughly syllable-like) http://juditacs.github.io/2019/02/19/bert-tokenization-stats.html
- VS stays with meaningful units, but this raises hard questions. Sometimes the meanings are very clear, Sanskrit *smi* 'smile', *vadh/badh* 'slay'; Hebrew *t.l.p.n* 'telephone' But often the meanings are more hazy, as in Skt *aNh* 'narrow, distressing', English *be* (am, are, is, was, were, would)
- What does *-er* as in *bigger, smaller* mean?

Semantics

- We have a large number of axioms (one for each word or morpheme, so on the order $10^4 - 10^5$)
- Given a sentence composed of a bunch of these, compute the meaning of the sentence
- If you rely on the manner of how these are comoposed together, linguistic semantics still considers this fair (e.g. for PP attachment)
- Dual picture: words are both formulas and vectors
- Both are legit. Vectors can be computed based of formulas alone (Ács, Nemeskey, and Recski, 2019)

Vector semantics

Up until this point, everything is prety standard. From here onwards, it is more about Kornai, 2023, which makes specific suggestions as to

1. The format of the axioms
2. The algorithm of putting words together
3. How this cashes out for word vectors
4. How the system addresses well known problems of linguistic semantics such as space, time, indexicals, negation, measure, quantification, deontic and epistemic modalites, etc.

Axioms

- These are called ‘meaning postulates’ in Montague grammar
- In the simplest case, just a conjunction of other words
  fox `animal, red, clever`
  wrong `lack right, avoid, hurt, lack correct, lack proper`
- Aims at *naive* worldview, not at scientific truth
- Monosemic
- Slightly circular (no, this doesn’t make it meaningless)
- Definition syntax is rigid (has a yacc/lex style)
- There are some 770 primitives in `4lang`, system comes with completeness guarantee (everything else can be defined in terms of these)

Relations

- Everything reduced to a dozen binary relations, AT, BETWEEN, CAUSE, ER, FOLLOW, FOR, FROM, HAS, IN, INS, ISA, LACK, MARK, ON, PARTOF, UNDER
- These have two arguments =agt, =pat
- No need for ternary predicates (this is a big deal for linguists)
- Binaries correspond to matrices, the rest are vectors
- Thought vectors are really matrices (situations)
- Easier to depict in (hypernode) graphs, can use RDF to linearize: *Brutus killed Caesar* `(Brutus cause (Caesar die))`
- Underlying logic is weak MSO (the only quantifier is gen) with defaults
- gen is just the vector $(1/d, 1/d, \ldots, 1/d)$

Conceptual structures

- Similar to classic Schankian scripts: a sequence of stills from a movie
- In the simplest case, we have just one still, which can include `before` and/or `after` clauses
- *-er* is just a comparison operator like '¿' (conceptual, but no visual image), cf. words like *betray*
- Contrary to expectations, children acquire conceptual/abstract words together with concrete words, not a moment later
- They offer substitution *salva veritate*

Thank You

Ács, Judit, Dávid Márk Nemeskey, and Gábor Recski (2019). "Building word embeddings from dictionary definitions". In: *K + K = 120: Papers dedicated to László Kálmán and András Kornai on the occasion of their 60th birthdays*. Ed. by Katalin MĂAdy BeĂAta Gyuris and GĂAbor Recski. Research Institute for Linguistics, Hungarian Academy of Sciences (RIL HAS).

Arora, Sanjeev et al. (2015). "Random Walks on Context Spaces: Towards an Explanation of the Mysteries of Semantic Word Embeddings". In: *arXiv:1502.03520v1* 4, pp. 385–399. DOI: 10.1162/tacl_a_00106.

Bengio, Yoshua et al. (2003). "A Neural Probabilistic Language Model". In: *Journal of Machine Learning Research* 3, pp. 1137–1155. DOI: 10.1162/tacl_a_00059. URL: http://www.jmlr.org/papers/v3/bengio03a.html.

Collobert, Ronan et al. (2011). "Natural Language Processing (Almost) from Scratch". In: *Journal of Machine Learning Research (JMLR)*.

Firth, John R. (1957). "A synopsis of linguistic theory". In: *Studies in linguistic analysis*. Blackwell, pp. 1–32.

Gittens, Alex, Dimitris Achlioptas, and Michael W. Mahoney (2017). "Skip-Gram – Zipf + Uniform = Vector Additivity". In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vancouver, Canada: Association for Computational Linguistics, pp. 69–76. DOI: 10.18653/v1/P17-1007. URL: http://aclweb.org/anthology/P17-1007.

Kiparsky, Paul (1982). "From cyclic phonology to lexical phonology". In: *The structure of phonological representations, I*. Ed. by H. van der Hulst and N. Smith. Dordrecht: Foris, pp. 131–175.

Kornai, András (2023). *Vector semantics*. Springer Verlag. DOI: 10.1007/978-981-19-5607-2. URL: http://kornai.com/Drafts/advsem.pdf.

Mikolov, Tomas, Wen-tau Yih, and Geoffrey Zweig (2013). "Linguistic Regularities in Continuous Space Word Representations". In: *Proceedings of the 2013 Conference of the*

North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2013). Atlanta, Georgia: Association for Computational Linguistics, pp. 746–751.

Pennington, Jeffrey, Richard Socher, and Christopher Manning (2014). "Glove: Global Vectors for Word Representation". In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, pp. 1532–1543. DOI: 10.3115/v1/D14-1162. URL: http://www.aclweb.org/anthology/D14-1162.

Schütze, Hinrich (1993). "Word Space". In: *Advances in Neural Information Processing Systems 5*. Ed. by SJ Hanson, JD Cowan, and CL Giles. Morgan Kaufmann, pp. 895–902.

Strobl, Lena et al. (2024). "What Formal Languages Can Transformers Express? A Survey". In: *Transactions of the Association for Computational Linguistics* 12, pp. 543–561. DOI: https://doi.org/10.1162/tacl.a.00663.