Episode Book

EPISODE #6398

Why Named Entity Linking Fails After the Span Is Found

PUBLISHED	1 November 2026 at 09:24 UTC
DURATION	26:25
LISTEN	myweirdprompts.com/episode/named-entity-linking-wikidata-22800/
HOSTS	Corn, Herman, Hilbert
SCRIPT MODEL	anthropic/claude-sonnet-5.5
VOICES	chatterbox-regular (chatterbox-local)
PIPELINE	V5.26
SOURCES	20 cited by the research (listed at the end)
BOOK GENERATED	2 November 2026 at 06:29 UTC

SYNOPSIS

Span detection across text is mostly solved, but resolving ambiguous entities remains an engineering hurdle. A look at the evolution of named entity recognition reveals the gap between classic rule-based annotators, supervised feature models, and modern frontier models. Examining tools like OpenTapioca, Solr lookups, and Wikidata aliases shows why generating candidates and disambiguating identities matters far more than just parsing words from a transcript.

TRANSCRIPT



Corn

...and the pipeline looked at Yachad, the Israeli political party, and decided with total confidence that it was a disability charity of the same name. Same spelling, different organisation.



Herman

It cleared that one in the enrichment pass, but it sat on the site as a fact for a while. That's what got me thinking about the whole stack behind it.



Corn

Daniel wants the full tour of named entity recognition. What the classic programs were, and how they differ from the modern ways of finding entities in text. The podcast now runs every episode through an entity step that pulls out people, organisations, places, works and events, links them to Wikidata, and builds an index on the website. His worry is the hard cluster: one name spelled five ways, aliases, transliterations, and two different things wearing the same name. Then, if you were building this today, which models and tools you'd actually reach for.



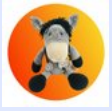
Herman

The pipeline already exists, so "what is NER" isn't the interesting question. Where does ours fail without telling anyone, and what would we build differently?



Corn

I'm in favour of anything that finds the failures before the listeners do.



Herman

Then definitions first, because the definition explains most of the difficulty. Herman Poppleberry, pedant, reporting for duty.



Corn

You've been waiting to say that.



Herman

Since breakfast. Named entity recognition is the information-extraction subtask of locating names in unstructured text and classifying them into predefined categories. Person, organisation, location, geopolitical entity, and also time expressions, quantities, monetary values, medical codes. The canonical example is "Jim bought three hundred shares of Acme Corp. in two thousand six." Jim is a person, Acme Corp. an organisation, two thousand six a time.



Corn

Textbook sentence. Nobody has ever bought exactly three hundred shares of anything.



Herman

The term itself comes from the sixth Message Understanding Conference, MUC-6, the evaluation campaign in the mid nineties. It covered ENAMEX, entity name expressions, so persons, locations and organisations, and NUMEX, the numerical ones.



Corn

Why "named" though? Why not just "entity"?



Herman

Because "named" is doing real philosophical work. It goes back to Saul Kripke and rigid designators. "Biden" and "the United States" are named entities because they pick out one specific object in every possible world. "President" is not, because in different worlds it picks out different people. So the task isn't arbitrary. It's about expressions that point at exactly one thing.



Corn

Which is the trouble, because the pointing is where it goes wrong. Yachad points at two things.



Herman

Right, and that gives you the three-step decomposition. NER finds the span. Named entity disambiguation decides which thing the span refers to. Named entity linking ties it to a record in a knowledge base, in our case a Wikidata QID. Recent work often does the first two jointly and calls it NERD.



Corn

Somebody was proud of that acronym.



Herman

My thesis is that span-finding is largely solved. You can run it on a laptop. The identity part is where a pipeline lives or dies.



Corn

Then let's go through the span part quickly, since it's solved.



Herman

Solved-ish. The history is worth a few minutes because it explains why the tools look the way they do. Three layers. First, rules, lexicons and ontologies. GATE's ANNIE is the canonical rule-based pipeline. Gazetteers of known names, patterns like capitalised word followed by "Inc.", grammar rules written by linguists.



Corn

So someone wrote down every way a company could end.



Herman

Pretty much. And the weakness is obvious the moment a company calls itself something unconventional. A pattern that expects "Inc." or "Ltd." has no idea what to do with a firm that's just named after a fruit. Stanford CoreNLP is the revealing case, because even the famous statistical tool was a hybrid. Its NER annotator uses machine-learning sequence models to label entities, but it also calls specialist rule-based components for times and dates, and there's a separate RegexNER annotator for patterns you just want to hand-write.



Corn

The machine learning tool that quietly uses regexes for dates. Sounds right.



Herman

Dates are a closed grammar, so rules are great at them. Nobody invents a new month. The second layer starts around two thousand seven, feature-engineered machine learning. The Stanford classifier, CRFClassifier, is a general implementation of linear-chain conditional random fields. You hand it features. Is the word capitalised, what's the suffix, what's the previous word, is it in a list. The model learns weights, and it labels the whole sequence at once rather than word by word.



Corn

Why did that beat rules?



Herman

It learned the soft patterns nobody could write down. But the survey literature is blunt that those models were domain-specific and did not generalise well to new data. Train on Wall Street Journal, point it at tweets, watch it fall over.



Corn

Which is why there's an OSU Twitter tagger at all.



Herman

Yes, a whole separate system because newswire models couldn't read tweets. Capitalisation is a great feature in a newspaper and nearly useless in a post written entirely in lowercase. The same era gave you Illinois NLP, NERsuite from academia, and in the open source and industry world LingPipe, NLTK, OpenNLP and spaCy. The third layer is deep learning over characters, sub-words and word embeddings. More robust to new domains, but it eats annotated data, and the survey calls that a bottleneck. AllenNLP's baseline is a nice snapshot. ELMo embeddings, GRU encoders over characters and phrases, GloVe vectors underneath, trained on CoNLL-2003.



Corn

I notice that sentence has four things in it I'd have to install.



Herman

You'd have had to install them. NeuroNER wrapped a neural tagger in something friendlier, but yes, this was not an era of one-line setup.



Corn

And all of it graded against the same few corpora.



Herman

The benchmarks defined what "good" meant. MUC-6 was three hundred and eighteen annotated Wall Street Journal articles. CoNLL-2003 has four types, persons, locations, organisations and miscellaneous. OntoNotes is richer. The New York Times corpus is over one point eight million articles, with around one point five million manually tagged. And in biomedicine, GENIA and NCBI-Disease. That last one is seven hundred and ninety-three PubMed abstracts and about seven thousand disease mentions mapping to just seven hundred and ninety concepts.



Corn

Ninety-one percent of mentions mapped to a single concept, I think. So even the hard medical set is mostly easy.



Herman

Mostly easy at the head. Keep that in your back pocket for later. Now compare the schemas. CoNLL's four types are baked into the model's output layer. Want "product" or "event"? Retrain, with new annotation.



Corn

Which is the entire problem with our index. Person, org, place, work, event, product. CoNLL has "miscellaneous" for the last three.



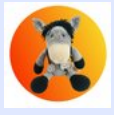
Herman

Miscellaneous was the junk drawer. Modern encoders let you pass arbitrary natural-language labels instead. We'll get to that. First, the shift to transformers, then zero-shot LLMs. This is where I want to be careful, because the marketing story is that you just ask a big model and you're done.



Corn

That's what we do, as it happens. DeepSeek reads the script and lists everybody.



Herman

And it works well enough to build a website on. But the literature is candid. The DiZiNER paper at ACL this year says plainly that zero-shot NER still lags far behind supervised systems. Their method has models disagree with each other, then refines the instructions where they disagree. It reaches zero-shot state of the art on fourteen of eighteen benchmarks and improves prior bests by eight F1 points.



Corn

And that closes the gap with supervised?



Herman

Reduces it by over eleven points. Not eliminates. The pattern across the papers is more machinery, not less. A Portuguese study found ensembles of local LLMs beat the individual models on four of five datasets. CMAS at the web conference uses four cooperating agents, partly because naive prompts overlook context and, quoting them, the indiscriminate use of task demonstrations severely misleads the model.



Corn

So the cure for a model that guesses is four models arguing.



Herman

Cheaper than hiring a linguist. Then there's the reasoning thread. ReasoningNER at AAAI argues that generative models excel at semantic pattern matching but lack an explicit, verifiable reasoning mechanism. They call it cognitive shortcutting. They train the model to reason through why a span is an entity, and report beating GPT-4 by twelve F1 points zero-shot.



Corn

Cognitive shortcutting. My entire approach to Monday mornings.



Herman

Then retrieval, which matters for us. NE-R1 targets long-tail, domain-specific entities where the model's own memory is thin, and retrieves on demand. The gains are modest, two and a half points in-domain and about one point cross-domain. But the direction is telling. When the entity is obscure, the model needs a lookup, not more parameters.



Corn

Herman, can I be the annoying one? Everything you've listed is a research result on a benchmark. What does a sensible person actually run for a fixed schema?



Herman

A small encoder. That's the honest comparison. Classic tools were brittle across domains but cheap and predictable. LLM approaches are flexible across schemas but need ensembles, reasoning scaffolds or retrieval to match a supervised tagger. And for a fixed set of types, a two-hundred-million-parameter model on a CPU can beat a frontier API call on cost and latency. Would you pay per episode to do something a laptop does?



Corn

I'd pay to never think about it. But I'd want to know when it was wrong, and a laptop tagger gives me a confidence score where a chatty LLM gives me a paragraph.



Herman

That is a better answer than the one I had prepared.



Corn

I have them occasionally. Usually after a nap.



Herman

Everything we just described finds spans. Our pipeline needs something else. It needs to know which Jim, which Acme, and that's a different problem with a different failure mode.



Corn

Hold on, before we go there. Our pipeline doesn't have a tagger. The LLM finds the span and guesses the Wikipedia title in one go. So it isn't doing candidate generation at all.



Herman

It is, just implicitly, from memory. The model guesses a title, we resolve it to a QID. If that fails, a Wikidata search returns five candidates and the model picks. Those are the three stages of entity linking, with the first stage hidden inside a model's recollection. Formally it's NER, candidate generation, where you select possible entries from the knowledge base, and disambiguation, where you choose the right one. The textbook example is Paris



Herman (continued)

the city versus Paris Hilton.



Corn

Linked, provisional, unlinked. Those are our three outcomes.



Herman

And the provisional bucket is the uneasy one. The model picks from five search results, which is better than a blind guess but depends entirely on what the search surfaced. Ivan: Badger, badger, nine. Forty one, seven. The second oven is cold. Repeat, forty one, seven. The rye is under the third shelf and the baker counts the loaves twice. The crumb remembers the naming. Nightjar.



Herman

Forty one... seven. Confirmed.



Corn

What on earth was that?



Herman

Candidate generation, and why the search surfacing the right five is the hard part. OpenTapioca is the reference lightweight Wikidata linker, trained from Wikidata only. Entity recognition runs through a SolrTextTagger analyzer, and candidates come from looking up whether the mention matches an entity label or alias in Wikidata, stored in a Solr collection.



Corn

That's the whole trick? A lookup table?



Herman

A very large, live lookup table. That's why I say the alias index is the game. Every alias someone adds to Wikidata improves the pipeline for free. Every missing alias is a silent failure, because the right entity never even makes the candidate list. The disambiguator can't pick what it was never shown.



Corn

So the variant spellings problem is mostly an alias problem.



Herman

Mostly. And I'd say, though I'm not sure of this, that variants are the minority case, a long tail of awkward names on top of a mass of easy ones.



Corn

Noted. Moving on to what it filters out.



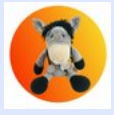
Herman

OpenTapioca keeps only person, location and organisation types. Our index wants works, events, products. A film, a conference, a phone model never even enters the net. That's a recall ceiling set by a type filter someone chose in two thousand nineteen.



Corn

Which is the argument against inheriting somebody's four categories.



Herman

Yes. There's also a smarter fix for retrieval than literal string matching. One paper indexes Wikidata's textual fields in Elasticsearch, then uses a seq2seq model to generate what the target entity's title and description should look like from the mention and its context, and queries with that. Instead of searching "Yachad", you search for something like "Israeli political party, centre-right, founded two thousand twenty-two".



Corn

So you describe the thing you're looking for, then look for the description.



Herman

Which dodges the same-string collision, at least partly. The charity and the party share a label, but their descriptions have almost nothing in common, so a query written from the episode's context lands near one and far from the other.



Corn

That assumes the context is in the sentence. Half our episodes say the name once and move on.



Herman

Then you pass a wider window, or the episode summary. Which is a real cost, and I won't pretend otherwise. Now transliteration, where I have to be straightforward. I don't know of a dedicated, widely-cited benchmark for Arabic, Chinese or Hebrew name variants into Wikidata. The adjacent evidence is suggestive, though. BioELX, an EMNLP Findings paper, finds that the aliases used to train cross-lingual biomedical linkers are heavily skewed toward English, so the retrievers generalise poorly to non-English mentions.



Corn

And naive fixes make it worse, I think you said.



Herman

They report that naively injecting context into retrievers trained only to align aliases severely degrades retrieval. Their fix is continued training on Wikidata-derived cross-lingual alias data, so concepts form shared neighbourhoods across languages, then an LLM reranks with the mention anchored in the prompt. Recall at one goes up by between five and eighteen points across four benchmarks.



Corn

And you speak Hebrew with an accent nobody can place. How do you spell Netanyahu?



Herman

Bureaucratically or colloquially? There are at least five standard romanisations of that name, and the one I use depends on who's reading. That's the problem. There isn't a single wrong spelling, so there isn't a single alias to add.



Corn

I'd have guessed this was the part the big models handle fine. They've read everything.



Herman

Head names, yes. For the tail, FiNERweb is a useful data point. Synthetic NER supervision across ninety-one languages and twenty-five scripts, and they find state-of-the-art models drop by two to nine hundredths of F1 when the labels are in the target language rather than English. Small, but the same phenomenon, English-shaped assumptions leaking in.



Corn

Two to nine hundredths. That's the sort of drop you'd only notice if you were looking.



Herman

Which is why it's the sort of drop that never gets looked at. Now disambiguation proper. Head entities, Paris and Apple, are easy. The population a niche podcast talks about is the tail. A study on long-tail entity linking in historical documents compared GPT and Llama three against a dedicated linker called ReLiK, on historical texts linked to Wikidata. They found LLMs perform encouragingly well, a valuable adjunct in filling the gap between head and tail. But they also note long-tail entities are underrepresented in training data and knowledge bases, and call the whole task understudied.



Corn

Encouragingly. That's researcher for "it mostly worked."



Herman

And "adjunct" is researcher for "not on its own". Which fits our Yachad case. The model had seen the charity more often than the party, so frequency won. Disambiguation needs a prior on how common something is, and context that overrides the prior.



Corn

So what would the override have looked like in that episode? The sentence around it was about Israeli coalition politics.



Herman

"Knesset", "coalition", "party list" sitting a few words away. A context-similarity score would have pulled hard toward the party. The model had the evidence in front of it and still reached for the more familiar name. That's the prior winning an argument it should have lost.



Corn

The episode about ASR benchmarks beyond word error rate comes to mind. A metric looked fine while the thing you cared about, whether the transcript was usable, was somewhere else.



Herman

Same shape. F1 on span-finding can look wonderful while the QID is wrong. Our equivalent of word error rate would be the share of entities that resolve to the right record, and I'd bet nobody's measured it.



Corn

Right, so, build it. If Daniel started from scratch tomorrow.



Herman

A stack, not a product. A fast transformer tagger for spans. Candidate generation against a Wikidata label and alias index, Solr or Elasticsearch as OpenTapioca does. Context-aware disambiguation, prior probability plus context similarity, or an LLM reranker. And an optional LLM pass for the long-tail cases that fall through.



Corn

And on the tagger, you'll tell me about GLiNER.



Herman

GLiNER2 from Fastino. One encoder around two hundred and five million parameters, BERT-based, doing NER, classification, structured extraction and relation extraction in a single forward pass. Apache licensed, CPU-first, no GPU required. And the interface is the part that matters. You call extract entities with the text and a list like company, person, product, location, in plain words.



Corn

So I describe what I want in English and it finds it. No retraining.



Herman

That's the schema-driven part. Compare that to a CoNLL tagger with four fixed outputs. If we decide tomorrow that we also want "podcast" or "book", we add a word to a list. It has over four million downloads, about four hundred and sixty thousand in the last month, and the family keeps growing. A large version at four hundred and eighty-six million, a two-point-five base at a hundred and ninety-four, a Decide variant, and a PII filter.



Corn

What's the catch? There's always one.



Herman

The catch is you've bought spans, not identity. It'll find "Yachad" beautifully and know nothing about which one. For linkers in the spaCy world, spacyopentapioca from Mannheim is the maintained one. Spacy-entity-linker and spacyfishing haven't been touched in years. Falcon two does joint entity and relation linking over Wikidata, and bbw from Mannheim adds typing and relations.



Corn

Which one would you actually pick?



Herman

For this show, I'd put GLiNER2 in front, a Wikidata alias index behind it, and keep the LLM only for the leftovers. The OpenTapioca README says its live server costs fifty euros a month. Linking is cheap. The language model is where the bill lives.



Corn

Fifty euros a month for the thing that took a decade of research. I spend more on leaves.



Herman

Do you want to know what worries me most about our current setup, though?



Corn

Go on.



Herman

The LLM guessing the Wikipedia title. If it's wrong, we get a confident, plausible, wrong link. If it's absent, we get a blank. Which would you rather debug?



Corn

The blank. A blank tells me to go and look.



Herman

A wrong link looks like a result.



Hilbert

Brother HL-5250DN. A hundred and sixty-nine dollars, two thousand eight. It's on a shelf in my hallway now, plugged into the same power strip as the router, and it's printing a list as we speak.



Hilbert

I compiled the surname index for a regional telephone directory then. Leeds patch. My job was counting the strays. That's what we called the spellings that wandered off from the proper one. Herman, you keep saying the tail. It wasn't a tail. Of the four thousand entries on my patch, six hundred and some had strays. One family alone had thirty-eight spellings in the book.



Hilbert

The list on the printer now is the entities your pipeline got wrong. I annotate it in pencil with what they should have been. I haven't entered a single correction anywhere. I want to see if it catches up by itself.



Herman

A hundred and sixty-nine dollars for a laser printer, and that's the best use anyone's had of it.



Corn

The pencil is the interesting part. That's a manual audit that runs on graphite.



Hilbert

Mine's been adding aliases to Wikidata too. One guest's name was spelled three ways across the episode pages and only one resolved. A short nickname for a local venue resolved cleanly to something else entirely. Nobody asked me to look.



Herman

Strays. I'm going to use that.



Corn

Six hundred out of four thousand is a lot of strays.



Hilbert

Eleven weeks, that count took me. One man on the team had it done in six and was wrong by a hundred and forty.



Herman

A hundred and forty off is exactly the kind of error that doesn't show up in an F1 score. And fifteen percent is a very different scale from the one I was assuming. If strays are that common, the alias index isn't a refinement. It's the main body of the work.



Corn

Which changes what you'd put on a dashboard. Not how many entities we found, but how many strays each one carried and who had checked.



Herman

And whether the alias lists we generate per entity, the ones our dedupe step builds, are ever pushed back anywhere or just sit in our own index.



Corn

Right now they sit. We take from Wikidata and give nothing back.



Herman

Which is the opposite of how the whole stack gets better. The linker improves when someone adds an alias, and the someone is going to have to be us, or a man with a printer.



Corn

The venue nickname is the one that stays with me. A wrong link that resolves cleanly is exactly the failure we were just calling worse than a blank.



Herman

And no confidence score would have flagged it, because the match was perfect. It was just a perfect match to the wrong thing.



Corn

The transliteration question still bothers me. If there isn't a benchmark, someone has to build one. Five romanisations of one name is a ready-made test set.



Herman

A test set where the labels are alias lists rather than single answers. That's the benchmark I'd want to see.



Corn

Then there's the tail, which is where a niche podcast lives. Our guests, our local venues, the odd political party. None of it is Paris.



Herman

The span part you can run on a CPU and mostly forget. The identity part is a standing relationship with a knowledge base, and the quality depends on how many strays somebody is willing to chase.



Corn

Eleven weeks of chasing, apparently, for four thousand names.



Herman

Our thanks to Hilbert Flumingtop for producing, and for keeping the whole thing running.



Corn

For more along these lines, there's episode ten, How ASR Went From Frustration To … Whisper Magic. This has been My Weird Prompts. If you enjoyed the episode, leave us a review wherever you listen, it helps more than you'd think.



Herman

We'll be back soon.

SOURCES

What the research for this episode read before the script was written. Primary sources first.

1. **Stanford CoreNLP NER docs . [primary]**
https://stanfordnlp.github.io/CoreNLP/ner.html

2. **GLiNER2 base model card, created 2025-07-02, last modified 2026-09-28 . [primary]**
https://huggingface.co/fastino/gliner2-base-v1

3. **OpenTapioca repo, Apache-2.0, last push 2026-08-26 . [primary]**
https://github.com/opentapioca/opentapioca

4. **Wikipedia, last edited 2026-08-07.**
https://en.wikipedia.org/wiki/Named-entity_recognition

5. **Wikipedia, last edited 2026-05-29.**
https://en.wikipedia.org/wiki/Entity_linking

6. **Wikipedia, last edited 2026-05-18.**
https://en.wikipedia.org/wiki/Named_entity

7. **A Brief History of Named Entity Recognition (survey performed 2020).**
https://arxiv.org/abs/2411.05057

8. **A Replicable Comparison Study of NER Software: StanfordNLP, NLTK, OpenNLP, SpaCy, Gate, SNAMS 2019.**
https://mpapad.github.io/publications/pdfs/SNAMS2019.pdf

9. **GLiNER2 paper (Zaratiana et al., 2025).**
https://arxiv.org/abs/2507.18546

10. **Delpeuch, OpenTapioca: Lightweight Entity Linking for Wikidata, 2019-04-19.**
https://arxiv.org/abs/1904.09131

11. **Survey on English Entity Linking on Wikidata: Datasets and approaches, Semantic Web.**
https://doi.org/10.3233/sw-212865

12. **Improving Candidate Retrieval with Entity Profile Generation for Wikidata Entity Linking.**
https://arxiv.org/pdf/2202.13404v3.pdf

13. **Evaluation of LLMs on Long-tail Entity Linking in Historical Documents, 2025-05-06.**
https://arxiv.org/abs/2505.03473v1

14. **BioELX, EMNLP Findings 2026, revised 2026-09-01.**
https://arxiv.org/abs/2605.27380v2

15. **DiZiNER, ACL 2026 Main, 2026-04-17.**
https://arxiv.org/abs/2604.15866v1

16. **A Reasoning Paradigm for Named Entity Recognition, AAAI 2026, 2025-11-15.**
https://arxiv.org/abs/2511.11978v1

17. **NE-R1, EMNLP 2026, 2026-09-02.**
https://arxiv.org/abs/2609.02366v1

18. **FiNERweb, 2025-12-15.**
https://arxiv.org/abs/2512.13884v1

19. **Local LLM Ensembles for Zero-shot Portuguese NER, 2025-12-10.**
https://arxiv.org/abs/2512.10043v1

20. **Cooperative Multi-Agent Framework for Zero-Shot NER, WWW 2025.**
https://arxiv.org/abs/2502.18702v1