Large Language Models and NER: better results with less work

Thalken, Rosamond Elizabeth
ret85@cornell.edu
Cornell University, United States of America

Wilkens, Matthew
wilkens@cornell.edu
Cornell University, United States of America

Mimno, David
mimno@cornell.edu
Cornell University, United States of America

Computational text analysis often hinges on identifying a key concept – a named entity, sentiment value, or writing style – in unformatted text. To identify these concepts, digital humanists tend to use rule-based or annotation-heavy methods, but the complexity and variability of humanities sources make authority lists and regular expressions insufficient. Commonly used named entity recognition (NER) tools from natural language processing, like Stanford NER, offer greater power but only recognize static categories of entities based on fixed collections of labeled data; they cannot easily adapt to unknown entity types or unfamiliar language patterns. These methods can lead to wasted effort and bias against data that does not fit within predetermined categories.

Our work considers how advances in neural language models might make NER more accurate, flexible, and streamlined for the digital humanist. We provide an example of how text-to-text generative models can identify mentions of characters, authors, and book names within Goodreads book reviews, and compare our results to other named entity recognizers. The methods from our immediate project will allow fellow researchers to better parse book reviews for entities of interest, but it will also demonstrate how text-to-text generative models might be used to creatively classify objects of interest in other humanist work.

Text-to-text generation is a recent advancement in language modeling and deep learning where a model takes a segment of text as input and outputs text without constraint (Raffel et al. 2020).

These large language models are pretrained on an enormous amount of text, but can be finetuned on a specific dataset and to learn a given task, like NER. Digital humanists have engaged with text generative models, especially GPT-3, to creatively generate poetry or imitate a writer's style (Elkins and Chun 2020, Hua and Raley 2020). Other work has used text generation to organize and describe narratives, such as identifying heroes and villains in plots or speeches (Stammbach et al. 2022).

The procedure we follow for generating a corpus-specific NER system takes only a few simple steps. We first create a spreadsheet with two columns: one for text input examples and the other for the output we want to generate for each example. Second, we "finetune" a pretrained language model using those example pairs. Finally, we use the finetuned model to annotate new examples.

We begin with an example in historical biodiversity literature. The inputs are paragraphs from botanical descriptions and the outputs are strings formatted to identify Latin names of plant species mentioned in the input text. We then finetune a generative language model (T5). A real example of input and generated text is in table 1.

In this biodiversity example, we found that not only was the model able to identify key entities, but it was also able to expand abbreviated names. In table 1, two plant names occur – with the same genus – making it possible to correctly resolve that the "C." in the second name is an abbreviation for "Chrisops." In book reviews, reviewers may reference a character by their nickname or an author by their last name, and given enough context, a generative text model could infer the full name, making it easier to organize and find information in reviews.

Table 1: Example of text generation for plant names.

Input Text (name bolded)	Generated Text
I took the following note when I saw the type in Genoa, a single specimen: "very like signifer Wk, only face altogether yellow; first abdominal segment yellow. May be only a paler variety **Chrysops dispar (Fab)**. I believe **C. impar Rond**.	Genus = Chrisops, Epithet = dispar, Author = Fab; Genus = Chrisops, Epithet = impar, Author = Rond

Although the output appears to be carefully formatted into named data fields, it is in fact generated by the model as an unconstrained string. The "Genus = ..." format is generated because we provided strings in this format during model finetuning. It is possible to train the model to produce any similar structured format *without requiring any additional coding*. In a book review, we can add categorization to clarify whether the reviewer is discussing an author or a character. The generated text would provide information about who is mentioned in the review, but we could use formatting categories to compare how entities are discussed within their higher-level categorical roles, like "author" or "character."

Compared to previous entity annotation systems, text-to-text generation systems offer more sophisticated results with less required technical skill. Researchers will be more able to combine their specialist knowledge of a collection with the generalization potential of language models trained on massive collections of text.

Bibliography

Elkins, Katherine / **Chun, Jon** (2020): "Can GPT-3 Pass a Writer's Turing Test?" in *Journal of Cultural Analytics* .

Hua, Minh / **Raley, Rita** (2020): "Playing With Unicorns: AI Dungeon and Citizen NLP , " *Digital Humanities Quarterly* .

Raffel, Colin / **Shazeer, Noam** / **Roberts, Adam** / **Lee, Katherine** / **Narang, Sharan** / **Matena, Michael** / **Zhou, Yanqi** / **Li, Wei** / **Liu, Peter J.** (2020): "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer," in *JMLR* .

Stammbach, Dominik / **Antoniak, Maria** / **Ash, Elliott** (2022): "Heroes, Villains, and Victims, and GPT-3: Automated Extraction of Character Roles Without Training Data." In *Proceedings of the 4th Workshop of Narrative Understanding* .