What's new in Computational Linguistics: an overview

Ilaria Salogni

August 31, 2024

Abstract

I wrote this text as a paper for the Digital Culture Seminar of the Digital Humanities faculty of the University of Pisa, taking the opportunity to deepen and complete in a personal research what is the linguistic representation within the LLM. To do so, I first talk about some huge changes that took place in Natural Language Processing in the past 10 years: the new role of corpora and of the classical pipeline, the introduction of embeddings.

This change was led by new technologies, first of all Transformers [Vaswani et al., 2017], which marked a real page turn and made this an even more exciting time for student, scholars, enthusiasts and everyone working with Language Modeling right now.

New results have also led to new questions, inter alia the ones about the role of linguistics and linguists in current language modeling.

I took this opportunity for a personal reworking, not only of what was covered in some seminars[1], but also basically all I've learned course after course during my Master's degree in Human Language Technologies, which gifted me the enthusiasm for the research in NLP, for which I am immensely grateful.

1 Introduction

In the first part, I will try to give an overview (far from exhaustive) of a few technical innovations and the huge changes that they represented in the recent-past of Computational Linguistics, aside the equally huge successes they led to in many tasks. This will be also necessary for crafting the speech in the second part.

We increasingly hear that the amazing recent technological advance in NLP has come at the price of interpretability. What does this really mean? What are the things we can't really explain, and what are the things that are understandable if we have the patience to delve into very technical and very specialized topics? To answer these questions, in the second part we will deal with what is the linguistic representation learned from the LLMs (Large Language Models). Furthermore, is linguistic theory still given a role within language modeling? If it is true that all linguists are left with is the enormous task of explainability, can probing tasks be trusted to assess the syntactic and semantic representation inside a black-box model?

[1] Fabrizio Sebastiani, 5 Ottobre 2022 "Supervised ML for the analysis and management of text: the ISTI-CNR experience"; Giuseppe Attardi, 2 Novembre 2022 "Utopie Digitali"

1.1 What is a LLM

The term Large Language Model (LLM) encompasses a broad range of models, including recurrent neural networks (RNNs), Transformers-based models, and various other neural architectures, all designed to process and generate human-like language. So, LLMs are by-definition based on more technologies than Transformers only, but Transformer-based models are the ones that took us all the way to ChatGPT-like performance. In really, really few words, in 2017 the 15-pages-long paper *Attention is all you need* [Vaswani et al., 2017] was first published. From there, NLP transitioned from deploying task-specific Machine Learning models (mainly RNNs) to Transformers-based models.

RNNs were originally designed for sequence tasks, and in a really low-level definition a sentence is as a matter of fact a sequence of words, but suffered from limited parallelization and struggled with capturing long-range dependencies. Transformers emerged with a new mechanism, called self-attention, enabling effective modeling of context within sequences: easier said than explained.

Attention is all you need: Again this was first introducted by [Vaswani et al., 2017] in the omonimous paper. Self-attention, a fundamental component in Language Models (LLMs), is a mechanism that enables the model to weigh the importance of different words or tokens within a sequence, allowing it to capture contextual relationships and dependencies effectively. The attention mechanism utilizes a set of learnable parameters to calculate attention scores for each word or token in a sentence, a mechanism that fits well out natural language speakers perception that not all words in a sentence contribute equally to its overall meaning.

Computationally speaking, Transformers eliminated the need for recurrent connections and introduced parallelization, resulting in faster training and inference. Computational-linguistically speaking, the Transformers architecture achieved remarkable success in various NLP tasks, both in notoriously hard ones, like machine translation language understanding and text generation, and in easy ones like NER, and there's no need for me to talk about the enthusiasm that came with these new Language Models.

Anyway, everything success comes with a price: unlike simpler models, where the relationship between inputs and outputs can be easily traced, Transformers-based models' output is derived from complex computations and interactions between numerous parameters and layers, making it challenging to understand how it arrives at its predictions. This is why they are called **black-box** models.

My aim for the first part is to focus on the following apparently simple sentence: *Transformers models' ability to capture global dependencies and handle large-scale data is what makes them a good choice in language modeling.*

1.2 Explainability vs Interpretability

Before facing on the sentence we have just stated, it is necessary to make a distinction, recently drawn in the field of LLMs: Explainability and Interpretability are not really the same thing [Alishahi et al., 2019].

Interpretability concerns the comprehensibility of the internal mechanisms and representations of the language model itself, the *what is going on inside*. It involves understanding how the model

processes and transforms the input data, revealing the relationships between different parts of the model and their impact on the output. Probing tasks are often used for interpretability, and we'll see some of those in the next pages. On the other hand, Explainability concernes the *why*, aims to uncover the reasoning or logic behind the model's output, allowing users to grasp why a particular prediction was made. Explainability often involves providing post-hoc explanations, such as generating textual or visual justifications that highlight the important features or evidence influencing the model's decision. When doing explainability, you want to understand why that was the output, given that the engine works like that.

Linguists possess deep knowledge of natural language structures, rules, and semantics. This doesn't necessary enables them to uncover the underlying linguistic principles behind the behavior of language models, or clearly delineating the linguistic abilities to be expected by such systems [Linzen, 2018], if there are any. Anyway they seem fit to explain to human language speakers, that only know their human speaker experience with language (that they have developed since birth together with their own conception of how human language works, without even knowing it and without ever thinking about it) what happens during computer-based processing of the language. This seems like a heavy task that requires linguistic and computer skills: therefore perfect for a computational linguist.

2 First part: to capture global dependencies and handle large-scale data

2.1 Corpora

> *One of us, as an undergraduate at Brown University, remembers the excitement of having access to the Brown Corpus, containing one million English words. Since then, our field has seen several notable corpora that are about 100 times larger, and in 2006, Google released a trillion-word corpus with frequency counts for all sequences up to five words long. In some ways this corpus is a step backwards from the Brown Corpus: it's taken from* ***unfiltered*** *Web pages and thus contains incomplete sentences, spelling errors, grammatical errors, and all sorts of other errors. It's* ***not annotated*** *with carefully hand-corrected part-of-speech tags. But the fact that it's* ***a million times larger*** *than the Brown Corpus outweighs these drawbacks. A trillion-word corpus—along with other Web-derived corpora of millions, billions, or trillions of links, videos, images, tables, and user interactions* ***captures even very rare aspects*** *of human behaviour.*[Halevy et al., 2009]

It would be difficult to explain better than the authors of *The Unreasonable Effectiveness of Data* [Halevy et al., 2009] the new relationship of the LLMs with textual data. The title is not random of course, but recalls to Eugene Wigner's article *The Unreasonable Effectiveness of Mathematics in the Natural Sciences* [Wigner, 1960], and the ability of mathematical explanations to capture and formalize complex phenomena, revealing deep underlying structures and patterns that may not be readily apparent through intuitive reasoning alone.

Corpora have always been the starting point of Computational Linguistics, there's no news in that. Stated this, everything else is different: if in the first models meticulously and fully annotated (on several levels) corpora by qualified scholars were required, the LLMs don't rely heavily anymore on manually annotated corpora leveraging unsupervised or self-supervised learning approaches. At

least they don't need them during the pre-training phase, while they do utilize labeled data for fine-tuning to improve their performance on specific tasks. This pre-training and fine-tuning paradigm is called **transfer learning**.

So, LLMS can get around the data scarcity problem and domain adaptation, becoming suitable for low-resource languages and other tasks with limited available data. This is no small thing, even if it comes together with the need for an immense amount of training data, and with many concerns about what may emerge from that data [Bender et al., 2021].

2.2 From words to sub-words

My very first NLP project in my first exam of CL with prof. Ježek, back in bachelor's degree, was to analyze the text of a book of choice using Python. My very first error was to use *.split()* so a tokenizer on whitespace, instead of nltk.word_tokenize(), a still rule-based but more sophisticated tokenizer, expecting good results.

Indulging in a little drama, we can say that NLP no longer deals with words. This is of course a stretch, and we can say it actually never worked on word-level, because even if rule-based tokenizers from the past relied on predefined linguistic rules to split text into tokens (the simplest of them all being that whitespace separates words) they still (should have) handled morphology, such as enclitic pronouns (some italian examples: *prendetelo, andatevene*). But if practice they often struggled with handling morphologically complex languages, out-of-vocabulary (OOV) words, and new words.

New **Sub-word tokenizers** use algorithms like Byte Pair Encoding (BPE) [Gage, 1994] or SentencePiece [Kudo and Richardson, 2018], instead of using a list of rules, starting with a vocabulary consisting of individual characters or a predefined set of tokens and iteratively merging the most frequently co-occurring sub-word pairs until a specified vocabulary size (or convergence) is reached. By breaking unseen words, sub-word tokenizers can handle morphological variations, Out-of-vocabulary words and even seamlessly handle multiple languages [Kaya Yiğit Bekir, 2022].

The **vocabulary size** is the only limit for the models' adaptation ability to different domains and languages, to which recent works have proposed a solution [Mofijul Islam et al., 2022]

2.3 Pipelines

Back to my first Computational Linguistic exam in 2020, the classical pipeline (sentence splitting, then tokenizing, then lemmatizing, then POS tagging, then dependency parsing ecc) was still very much alive. In subsequent exams I was asked again to actually implement it, but I think more for educational purposes and because it was a fundamental step for Computational Linguistics, which certainly hasn't totally fallen into disuse today. For example, it may still be needed for very specific academic linguistic research tasks.

Anyway, while there is ongoing debate regarding whether Transformers models implicitly capture tree parsing structures, as we'll discuss later, it is clear that the classical pipeline approach in Natural Language Processing (NLP) has evolved significantly.

The traditional pipeline consisted of distinct stages to be performed one after the other, such as tokenization, part-of-speech tagging, syntactic parsing, and semantic analysis. Thanks to large

scale pre-training, Transformers models obviate the need for explicit syntactic parsing stages in the pipeline, encoding a rich linguistic representation directly from raw text. For example, the fact that these models work with text **sequences** (that vary from long paragraph to whole documents) and that updated the notion of sentence, eliminates the need of sentence splitting.

Another example is POS tagging that, while being still a thing, has been limited to situations that require granular linguistic analysis, like multilingual NLP and domain-specific linguistic analysis.

This reduces the need of handcrafted linguistic features and sequential processing steps, offering a more unified and end-to-end approach to NLP tasks (in few words, no inter-annotator agreement anymore). Explicit parsing stages have been overtaken by a new holistic paradigm: the embeddings.

2.4 Encoding into embeddings

if we take the vectors of a co-occurrence matrix ("dog" appears near "run" 5 times, near "sleep" 2 times and so on for every word in the text) and use them as distributional vectors, then we would be adopting an explicit distributional representation. Each individual vector dimension corresponds to a context. In this case, dimension = context. These vectors are higly dimensional and sparse. Yuck.

Implict vectors (embeddings) are low dimensional, dense, and their dimension are not the same as context anymore, but are the latent features, extracted from data. The negative side is that we loose interpretability. So, if embeddings are vectors, aka arrays of integer numbers, they look like this: [102, 504, 1818, ...]. What does those number represent?

Both Word2vec [Mikolov et al., 2013] and contextual embeddings are are dense vector representations, that are a broader group, opposed to sparse vectors.

Word2vec embeddings are static representations that assign a fixed vector to each word based on its co-occurrence statistics in a large corpus. These embeddings capture semantic relationships between words but lack the ability to consider the context in which a word appears. As a result, word2vec embeddings treat each occurrence of a word (type word) in the same way, regardless of its surrounding context. Every occurrence of "dog" in my text, will have the same embedding vector, and the length of this vector is an hyperparameter.

In contrast, **contextual embeddings** [Liu et al., 2020] capture the meaning of a word based on its context within a specific sentence or sequence. These embeddings are computed by taking into account the surrounding words and their positions, allowing the model to capture nuanced and context-dependent word representations [Mohebbi et al., 2023a]. So, every occurrence of "dog" will have a different embedding vector. Their legth is an hyperparameter also, but usually is fixed to 768 in BERT base. Contextual embeddings enable models to understand polysemous words with multiple meanings and disambiguate them based on the surrounding context. A big goal for mr. Firth[2] [Firth, 1957].

Our goal in this paragraph is just to state the strong empirical performance of contextualized word representation: these embeddings enable Transformers models to capture and understand the relationships between words that are far apart in a sequence. Although there still is a maximum

[2] *"You shall know a word by the company it keeps"*

length for input sequence (it was 512 tokens for BERT original model) due to computational constraints and memory limitations, that context is enough to learn 80 percent longer dependencies than RNNs [Dai et al., 2019].

Let's go back and recap before moving on: If we take a BERT model and feed it with our text, for each token we'll get his embedding that is an array-shaped object like [102, 5, 1909, ...], that like we said above represents our token in a 768-dimensional space, that encodes both local and global context, capturing dependencies between neighboring tokens as well as long-range relationships within the sequence. How can that make sense?

Tenney and colleagues [Tenney et al., 2019b] pose some better posed questions:

1. What information is encoded at each position and how well it encodes structural information about the word's role?
2. Is the encoded info syntactic or also from an higher level (semantic)?

These questions are going to take us to the second part of the work.

3 Second part: Linguistic representation in LLMs

3.1 Linguistics meets LLMs

Setting up a strict dichotomy between a linguistic theory and Language Models may not be necessary or productive: while a linguistic theory provides a structured understanding of language, LLMs offer powerful computational tools for processing vast amounts of text data, and we can settle for this. Of course language technologies perform better than syntactic representation based on introspection by trained linguists [Hill, 2023] , and I think linguists themselves would be the first one to wish to get rid of a theory that does not originate from data, and that is not quantitatively provable. How Halevy, Norvig et al. [Halevy et al., 2009] said in a 2009 work, *we should stop acting as if our goal is to author extremely elegant theories, and instead embrace complexity and make use of the best ally we have: the unreasonable effectiveness of data.* This position is fascinating for its balance, but is also easier said than done. Although *Transformers are obviously good language models* [Hill, 2023], for their great generalization power, they lack parsimony[3] [Wiechmann et al., 2013] and we cannot extract their internal representation to comment on it and make a linguistic theory out of it.

Our aim here is to understand the linguistic representation encoded by LLMs: while linguistic theory aims to explain how language works by proposing generalizations about the structure and meaning of language, probing tasks have emerged as a way to explain how model works by proposing a classification task that probes the structure of the model, and finally to infer a model's knowledge of linguistic properties.

[3]The principle that the most acceptable explanation of an occurrence, phenomenon, or event is the simplest, involving the fewest entities, assumptions, or changes (Oxford Reference)

3.2 The art of probing

With **probing tasks** we mean the various set of tasks that are used to try to probe precisely the lexical and semantic representation of the model. They are our tools to probe the internal layers of the model and assess the model's understanding of various linguistic phenomena. More tecnhnically, probing mean (usually) training a shallow supervised classifier that attempts to predict specific linguistic properties or reasoning abilities, based on representations obtained from the model. We have no gold standard of what happens inside a model [Mohebbi et al., 2023b], so we design a specific ad-hoc task for which a supervised setting is a viable option.

Even if reusability is always a goal, researcher usually goes on to designate a new and specific probing task for his or her research. This approach makes it difficult to observe consistency over more than a setting or a model, as probing techniques depend on the specifics of and encoder architecture [Conneau et al., 2018]. Once we have found the answer for a specific model, we can not expect to be able to explain the representation of all the other models [Fayyaz et al., 2021]. More than once, probing tasks have been found lacking of conclusive evidence in following works.

The following questions also are to be posed, in order to pursue scientificity in our job:

- Can we say that when a probe achieves high accuracy on a linguistic task using a representation, can we conclude that the representation encodes linguistic structure, or has the probe just learned the task? [Hewitt and Liang, 2019]
- Furthermore, most probing studies use linguistics as a theoretical scaffolding, and inevitably commits to the specific theoretical framework used to produce the underlying

  data [Kuznetsov and Gurevych, 2020].
- Finally, when probing, a researcher chooses a linguistic task and trains a supervised model, doesn't choose a linguistic theory: is having a clear research question in mind enough when setting up a probing task with scientificity claims, or is something at least resembling a linguistic theory needed?

It is still not in my power to answer fully these questions, that I will be more than happy to keep researching in the future, but we can see some examples of their practical implications in the next paragraph.

3.3 Probing the syntactic knwoledge

Now that we have introduced the concept of probing we can move on to some examples: we can ask ourselves, for example, how syntactic information is treated in the LLMs. Many probing tasks has been set up for trying to assess also how the linguistic information is treated and stratifies in each layer.

Jawahar [Jawahar et al., 2019] quantify their claim that *BERT mostly captures phrase-level information in the lower layers and that this information gets gradually diluted in higher layers* using a k-means clustering probing task, and in fewer words that *BERT embeds a rich hierarchy of linguistic signals: surface information at the bottom, syntactic information in the middle, semantic information at the top.*

Lin [Lin et al., 2019] agree on the fact that *BERT encodes positional information about word tokens well on its lower layers, but switches to a hierarchically-oriented encoding on higher layers.* But someone else disagrees with this vision, like Niu [Niu et al., 2022] that anyway can find a common ground on the fact that *BERT's structure is, however, linguistically founded, although perhaps in a way that is more nuanced than can be explained by layers alone.*

Changing just slightly our point of view and getting more specific, we may want to know whether syntax trees are embedded implicitly in deep models' vector geometry [Hewitt and Manning, 2019]. Reformulating in clearer terms this is equivalent to saying *BERT representations capture linguistic information in a compositional way, that mimics classical, tree-like structures* [Tenney et al., 2019a] or in a even stronger statement that BERT rediscovers the classical pipeline. Again Niu [Niu et al., 2022] claims that the probing tasks set up by Tenney [Tenney et al., 2019a] and [Jawahar et al., 2019] lack conclusive empirical support.

attention heads analysis Probing tasks can't do all the work themselves: additional techniques, such as analysis of attention patterns, can help. Attention heads, like we said before in 1.1, refer to the attention mechanisms used by language models to weigh the importance of different input tokens to generate their output representations. Analysis of attention heads involves identifying patterns in the attention weights assigned by specific attention heads, such as the degree to which they attend to certain parts of the input and how they interact with other attention heads. Can we hope by analyzing the attention heads to bring out if not really a linguistic theory at least some information on how syntax and semantics are treated by an LLM?

Clark [Clark et al., 2019] propose an attention-based probing classifier with this purpose, inter alia. The debate on the explanatory potential of attention heads is still at the beginning, and researcher are calling for more integration from different areas to get to an answer [Bibal et al., 2022].

So for now we are far away from explaining where how or if the syntactic info is stored in representations learned by LLMs. Transformers are great at syntax, because they operate precisely by inferring patterns between input word (pieces) in a way that optimally satisfies their objectives [Hill, 2023], a way that we can't really explain or use for our linguistic cravings.

3.4 Probing semantic knowledge

Our second question in 2.4 was: is the information encoded by a language model syntactic or also from an higher level, that is to say semantic? Again, it is difficult to infer whether the relevant information is encoded within the span of interest or rather inferred from diffuse information elsewhere in the sentence [Tenney et al., 2019b]; Of course, the idea of modeling sentence or context-level semantics together with word-level semantics proved to be a powerful innovation [Wiedemann et al., 2019] But again this came at the price of interpretability: if we want to capture the meaning of the sentence, we have to move away from the meaning of the word: to decode extra-sentential content, we need too contextual vectors [Hill, 2023].

This is to say that the answers to how is syntax encoded and how is semantic encoded are tied up in the same tangle.

4 Conclusion

Like I said in the introduction, this overview is far from exhaustive: I missed to talk about Natural Language Inference and how the pragmatic level is embedded in the representation, and I didn't say anything about the peculiar field of Machine Translation. There's so much to say also about how can LLMs deal with multiple languages at the same time. Much more can of course be said about this, and I am eager to dive deeper in these subjects in the future. I hope all the unanswered questions will make the reader feel curious and fascinated about what the future is going to carry like I am.

References

[Alishahi et al., 2019] Alishahi, A., Chrupala, G., and Linzen, T. (2019). Analyzing and interpreting neural networks for NLP: A report on the first blackboxnlp workshop. *CoRR*, abs/1904.04063.

[Bender et al., 2021] Bender, E. M., Gebru, T., McMillan-Major, A., and Shmitchell, S. (2021). On the dangers of stochastic parrots: Can language models be too big? . In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21, page 610–623, New York, NY, USA. Association for Computing Machinery.

[Bibal et al., 2022] Bibal, A., Cardon, R., Alfter, D., Wilkens, R., Wang, X., François, T., and Watrin, P. (2022). Is attention explanation? an introduction to the debate. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3889–3900, Dublin, Ireland. Association for Computational Linguistics.

[Clark et al., 2019] Clark, K., Khandelwal, U., Levy, O., and Manning, C. D. (2019). What does bert look at? an analysis of bert's attention.

[Conneau et al., 2018] Conneau, A., Kruszewski, G., Lample, G., Barrault, L., and Baroni, M. (2018). What you can cram into a single $&!#* vector: Probing sentence embeddings for linguistic properties. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2126–2136, Melbourne, Australia. Association for Computational Linguistics.

[Dai et al., 2019] Dai, Z., Yang, Z., Yang, Y., Carbonell, J., Le, Q. V., and Salakhutdinov, R. (2019). Transformer-xl: Attentive language models beyond a fixed-length context.

[Fayyaz et al., 2021] Fayyaz, M., Aghazadeh, E., Modarressi, A., Mohebbi, H., and Pilehvar, M. T. (2021). Not all models localize linguistic knowledge in the same place: A layer-wise probing on BERToids' representations. In *Proceedings of the Fourth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 375–388, Punta Cana, Dominican Republic. Association for Computational Linguistics.

[Firth, 1957] Firth, J. R. (1957). A synopsis of linguistic theory 1930-55. 1952-59:1–32.

[Gage, 1994] Gage, P. (1994). A new algorithm for data compression. *C Users J.*, 12(2):23–38.

[Halevy et al., 2009] Halevy, A., Norvig, P., and Pereira, F. (2009). The unreasonable effectiveness of data. *IEEE Intelligent Systems*, 24(2):8–12.

[Hewitt and Liang, 2019] Hewitt, J. and Liang, P. (2019). Designing and interpreting probes with control tasks.

[Hewitt and Manning, 2019] Hewitt, J. and Manning, C. D. (2019). A structural probe for finding syntax in word representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4129–4138, Minneapolis, Minnesota. Association for Computational Linguistics.

[Hill, 2023] Hill, F. (2023). Why transformers are obviously good models of language.

[Jawahar et al., 2019] Jawahar, G., Sagot, B., and Seddah, D. (2019). What does BERT learn about the structure of language? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3651–3657, Florence, Italy. Association for Computational Linguistics.

[Kaya Yiğit Bekir, 2022] Kaya Yiğit Bekir, T. A. C. (2022). Finding the optimal vocabulary size for turkish named entity recognition.

[Kudo and Richardson, 2018] Kudo, T. and Richardson, J. (2018). Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing.

[Kuznetsov and Gurevych, 2020] Kuznetsov, I. and Gurevych, I. (2020). A matter of framing: The impact of linguistic formalism on probing results.

[Lin et al., 2019] Lin, Y., Tan, Y. C., and Frank, R. (2019). Open sesame: Getting inside BERT's linguistic knowledge. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 241–253, Florence, Italy. Association for Computational Linguistics.

[Linzen, 2018] Linzen, T. (2018). What can linguistics and deep learning contribute to each other?

[Liu et al., 2020] Liu, Q., Kusner, M. J., and Blunsom, P. (2020). A survey on contextual embeddings.

[Mikolov et al., 2013] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In Burges, C., Bottou, L., Welling, M., Ghahramani, Z., and Weinberger, K., editors, *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc.

[Mofijul Islam et al., 2022] Mofijul Islam, M., Aguilar, G., Ponnusamy, P., Solomon Mathialagan, C., Ma, C., and Guo, C. (2022). A vocabulary-free multilingual neural tokenizer for end-to-end task learning. In *Proceedings of the 7th Workshop on Representation Learning for NLP*, pages 91–99, Dublin, Ireland. Association for Computational Linguistics.

[Mohebbi et al., 2023a] Mohebbi, H., Zuidema, W., Chrupała, G., and Alishahi, A. (2023a). Quantifying context mixing in transformers. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 3378–3400, Dubrovnik, Croatia. Association for Computational Linguistics.

[Mohebbi et al., 2023b] Mohebbi, H., Zuidema, W., Chrupała, G., and Alishahi, A. (2023b). Quantifying context mixing in transformers. In *In Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 3378–3400. Association for Computational Linguistics.

[Niu et al., 2022] Niu, J., Lu, W., and Penn, G. (2022). Does BERT rediscover a classical NLP pipeline? In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 3143–3153, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

[Tenney et al., 2019a] Tenney, I., Das, D., and Pavlick, E. (2019a). BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601, Florence, Italy. Association for Computational Linguistics.

[Tenney et al., 2019b] Tenney, I., Xia, P., Chen, B., Wang, A., Poliak, A., McCoy, R. T., Kim, N., Durme, B. V., Bowman, S., Das, D., and Pavlick, E. (2019b). What do you learn from context? probing for sentence structure in contextualized word representations. In *International Conference on Learning Representations*.

[Vaswani et al., 2017] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. *CoRR*, abs/1706.03762.

[Wiechmann et al., 2013] Wiechmann, D., Kerz, E., Snider, N., and Jaeger, T. F. (2013). Introduction to the special issue: Parsimony and redundancy in models of language. *Language and speech*, 56:257–64.

[Wiedemann et al., 2019] Wiedemann, G., Remus, S., Chawla, A., and Biemann, C. (2019). Does bert make any sense? interpretable word sense disambiguation with contextualized embeddings.

[Wigner, 1960] Wigner, E. (1960). The unreasonable effectiveness of mathematics in the natural sciences. *Commun. Pure Appl. Math.*, 13(1):1–14.