Research Statement

Antoine Simoulin

My primary research interest lies in the area of natural language processing (NLP) where I work on architectures of computational models to produce sentence embeddings. Building standalone sentence embeddings is specifically hard, as an infinite number of valid sentences exist. However, compositional semantics state that the meaning of a phrase is determined by combining the meanings of its subphrases, using rules. Models, therefore, need to compose text units, given a syntactic structure, into global semantic embeddings. Enabled by my lab's interdisciplinary research in both computational and theoretical linguistics, I include linguistic biases into neural networks. I then analyze how their inner sequence of compositions compare with linguistic theory and the gain enabled by such biases. This aspect of my work is detailed in section §1. I am also motivated by academics and industry applications of sentence embeddings, such as search engines or text mining. As detailed in §2, I train such models at scale to obtain state-of-the-art results in the domain of sentence embeddings and language modeling. I share a large portion of this work as open-source contributions, ready to use for real-world applications. In section §3, I detail how I intend to pursue my research to increase the model's controllability and intelligibility. I aim at integrating other kinds of biases into machine learning models such as combining symbolic and statistic approaches.

1. Toward integrating linguistic biases into neural networks

My first line of research aims at improving the compositional properties of machine learning models and their ability to generalize outside their training domain. I aim to integrate the recursive property of language within neural models. I design and analyze architectures based on linguistic theory.

Jointly learning model structure and compositional operations I focus on tree-structured neural networks, which naturally encode the structure of language. For each sentence, the network computes text units following a syntactic tree, starting from the leaf nodes, up to the root. However, such models suffer from practical constraints that limit their application. In particular, tree-based models not only require raw text as input but also the sentence structure in the form of a parse tree. Such structure may be tedious to obtain as it requires manual annotations and external parsers. To overcome such limitations, I formulated a novel tree-based model that learns its composition function together with its structure **[1]**. The model includes two modules, a biaffine graph parser, and a Tree-LSTM. The parsing and the composition functions are explicitly connected and, therefore, learned jointly. The method differs from previous work as the representation is not computed from the whole forest of potential trees. Moreover, training the full model directly does not require supervision from a parsing objective. The model outperforms tree-based models relying on external parsers on downstream tasks. In some configurations, it is even competitive with BERT-base model.

Studying shallow structure in transformer models Recent transformer architectures have gained increased popularity within the community. Contrary to tree-based models, they do not need carefully hand-annotated data to be trained. On the other hand, as many results suggest, these new models acquire some sort of tree structure. Transformers update each token hidden simultaneously through a fixed number of layers. Yet the role of these layers and how they process information is not fully understood. I formulate the hypothesis that the distinct layers do not encode specific surface, syntactic nor semantic functions but rather that such information emerges through the iterative application of layers. To better study the transformation of token representations across layers, I proposed a variant of ALBERT **[2]**. This model implements the key specificity of weights tying across layers, but also dynamically adapts the number of layers applied to each token. I analyze token transformation across the network depth. In particular, I study how iterations are distributed given the token dependency types. I showed that tokens do not require the same amount of iterations and that difficult or crucial tokens for the task are subject to more iterations.

Characterizing compositional properties of neural architectures While transformers show outstanding performances on many NLP benchmarks, they also have some linguistic limitations. In particular, regarding their ability to generalize outside their training range and to learn elementary composition rules. The benchmark COGS **[3]** for example highlights deep learning models struggle to generalize to longer sequences or sentences with deeper level of recursion than seen during training. Following my work on integrating structure into neural architecture, I aim at better characterizing how the model structure may affect their degree of compositionality. This work is currently in an experimentation phase. I am building an evaluation setup with arithmetic expressions containing specific properties. I train various models on specific subsets and observe how models generalize outside their domain. In particular, I compare models relying on different degrees of structure constraints such as sequential, recursive, or unstructured models.

2. Training language models at scale

My second line of research focuses on training and sharing models at scale. Indeed, the preparation of massive corpora, the training, and the use of large architectures are key for the performance of such models. Moreover, specific behaviors and linguistic properties deeply depend on the scale.

Training sentence embedding models using a discriminative objective Inspired from linguistic insights, I assume structure is crucial to building consistent representations. I indeed expect sentence meaning to be a function of both syntax and semantic aspects. In that regard, I proposed a self-supervised method that builds sentence embeddings from the combination of diverse explicit syntactic structures of a sentence **[4]**. The novelty consists in jointly learning structured models in a contrastive multi-view framework that induces an explicit interaction between models during the training phase. I pre-trained various models using a contrastive objective with a 40 million sentences corpus. I then evaluate my models on sentence embedding benchmarks and obtain state-of-the-art results. In particular on tasks that are expected, by hypothesis, to be more sensitive to sentence structure. From a practical point of view, implementing tree-structured models can be hard. I open-sourced the code I developed for recursive models under a library called PyTree[1]. The library was distinguished and listed among the winners of the PyTorch Hackathon 2021. Motivated to share state-of-the-art models, I also participated in a hackathon[2] to develop, train and release large sentence embeddings models. We used a similar contrastive objective and trained models on a 1 billion sentences corpora. We developed specific evaluation benchmarks for sentence embeddings and obtained state-of-the-art results. Our project was among the winners of the competition and received an honorable mention.

Training the first large language model for French using a generative objective As observed in **[5]**, deep neural networks have shocking grammatical competencies. For example, GPT-2 generates correct text with plural and long-distance agreement despite any prior linguistic knowledge. Such agreements are determined by abstract structures and not just linear order of words. Surprisingly, models can learn such specific linguistic patterns (subject-verb, noun-adverb, verb-verb) with no prior information about linguistic theory. Within my laboratory, I led the project to train the first large language model in French **[6]**. We obtained a dedicated computation grant on public French HPC computer Jean Zay. The model, equivalent to GPT-2 in English, contains more than 1 billion parameters. We built a dedicated training corpus and parallelized the training between multiple nodes and compute units. I am particularly proud of this project, as we contributed to the resources available in French. We released the model in Open-Source for research and business application purposes[3].

---

[1] https://github.com/AntoineSimoulin/pytree

[2] https://discuss.huggingface.co/t/open-to-the-community-community-week-using-jax-flax-for-nlp-cv/7104

[3] https://huggingface.co/asi/gpt-fr-cased-base

3. Future Research Directions

In my opinion, recent advances in NLP have opened up new and exciting applications. Yet, current architectures lack control and intelligibility properties. When using models for real-world applications, it is hard to avoid, let alone explain, unwanted behavior. In that perspective, I intend to pursue my research in the direction of integrating linguistic or formal theory into machine learning models and training such architectures at scale. In the following sections, I identify two main directions that may be of particular interest to increase the models' robustness to generalization outside their domains and provide efficient tools for improving intelligibility and control over language models.

Integrating symbolic and logic bias into language models Symbolic AI typically encodes knowledge using explicit rules. These systems may require extensive feature engineering to describe individual elements, but they are very effective at explaining how to compose them. By hard integrating composition rules, they are naturally more resilient to out-of-domain generalization. Combining symbolic systems and deep learning representation methods is an active subject of research. For example, to combine object recognition and reasoning abilities using generation of symbolic programs or by integrating logic into neural networks. Following my work to integrate structure constraints in neural networks, I aim to integrate logic constraints into architectures. In my opinion, such approach complements methods for intelligibility in deep neural networks. Indeed, we do not attempt to explain models afterward but rather try to constrain their architectures to provide more explicit or readable transformation sequences. Such approach may also enhance models out of domain generalization properties by providing new regularization methods.

Toward controllable text generation Language models currently integrate knowledge within the network hidden states. For natural language generation, we may observe unwanted behavior such as hallucination or factually incorrect statements. While such models may be used in original setups such as few-shots or zero-shot learning, they still lack controllable properties. Indeed, the generated statements depend on the architecture, the data used for pre-training, or the prompt used during the task. Many works focus on prompt engineering to control the model afterward. But we may also consider refining the architecture, as stated previously, or the form of the data used to infuse knowledge into the network. For example by using specific memory structures or by using conditional variables. Better controlling language models' generative properties may also help us reduce their exponentially growing size by identifying redundant parameters. In general, such properties appear critical to building consistent and robust systems for real-world applications.

References

[1] Antoine Simoulin, Benoît Crabbé: **Unifying parsing and tree-structured models for generating sentence semantic representations.** CoRR abs (2021)

[2] Antoine Simoulin, Benoît Crabbé: **How Many Layers and Why? An Analysis of the Model Depth in Transformers.** ACL (student) 2021: 221-228

[3] Najoung Kim, Tal Linzen: **COGS: A Compositional Generalization Challenge Based on Semantic Interpretation.** EMNLP (1) 2020: 9087-9105

[4] Antoine Simoulin, Benoît Crabbé: **Contrasting distinct structured views to learn sentence embeddings.** EACL (Student Research Workshop) 2021: 71-79

[5] Tal Linzen, Marco Baroni: **Syntactic Structure from Deep Learning.** CoRR abs/2004.10827 (2020)

[6] Antoine Simoulin, Benoît Crabbé: **Generative Pre-trained Transformer in_____ French.** TALN (1) 2021: 246-255