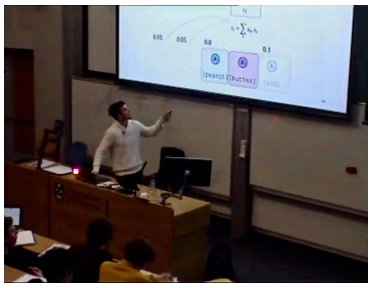
Carlos Gemmell Górriz

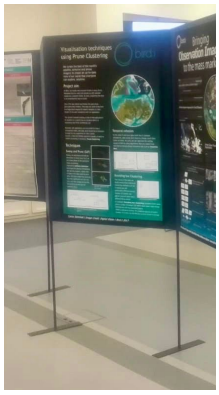
Address: 2/1, 21 Gibson St, G12 8NU, Glasgow | Phone: +44 07769389115 | Email: carlos.gemmell@gla.ac.uk

October 2019 - now: PhD at the University of Glasgow



I am a PhD student at the **GRILL Lab** in the department of Computing Science at the University of Glasgow (UofG) where I research the next generation of **deep neural architectures** for **program synthesis** and **conversational modeling** under the supervision of Jeff Dalton. My fundamental motivation is to **build agents capable of deep collaboration** to achieve complex goals through conversation to enhance rather than substitute humans. Currently, my work focuses on extending the capabilities of **Transformer architectures** for structure dependent tasks like **code generation** and improving its long sequence performance with **dense graph memory**. I work mainly in **PyTorch**, with experience in **Tensorflow 2.0**, and do daily experiments on available **multi-GPU** clusters and starting **TPU** training.
I am proud to have been published at **SIGIR 2020** (21% acceptance rate) for work integrating **information retrieval** (IR) and **Sequence-2-Sequence models for program synthesis**: Relevance Transformer, Gemmell et al. as well as speaker positions at **TREC 2020** for **Conversational Assistance**.
I have also assisted in **teaching** the **foundational NLP course at the UofG** giving lectures to a growing class of **300+ students.** I go deeper into NLP with additional workshops with like minded students.

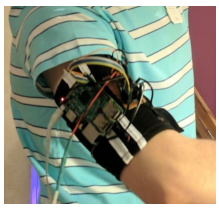
2015 - 2019: Bachelor in Computing Science at the UofG and Internships



I graduated with **First Class Honours** in **Computing Science** where, in my last year, I focused on **Algorithmics** and **Artificial Intelligence** and defended my thesis on using **deep recurrent networks for language conditioned code generation**.
Along the way I squeezed in **two summer internships** at Birdi, an **AI Startup** in satellite imagery where I worked in a great team led by Sivakumaran S. as a **research scientist in computer vision** helping **push ML models to production**. The research also included high performance data visualizations for customer click data leading to a novel sweep line pruning algorithm that was presented at UCL.

Projects outside of research and teaching

Hackathons: cyber defence and game design
First place at CDX 2017 among 30 teams for defending infected linux servers against attackers. First place at the 2018 UofG hackathon making a Unity 3D multiplayer online game in 48h.



Haptic Sense: a haptic interface between brain and machine
A wireless transmitter for plain text that is fed into the body through an array of 6 haptic motors. The array creates a pattern in braille mapping ASCII characters. **Think reading text through vibrations in your arm**. We achieved a speed of 4 characters per second.

ElectricGT Tesla LED Panel: car roof video display for an electric race series
EGT is an electric race series for which I **designed**, **prototyped** and **shipped** a large LED display to show live metrics of the race car while driving. It is made using 2400 LEDs and interfaced with a custom wireless JavaScript library. The panel was sold and installed in the car in 2017 during my second year at university.

Details about the PhD

Advances in Artificial Agents are showing an upwards trend for surpassing humans at many tasks. Medical diagnosis and prediction of judicial decisions are just a few areas where model perceptive capacity for detail, and pattern recognition, put in question our own capabilities. However, to set and achieve complex goals, perceptive capacity is insufficient and is where a collaboration fusing the strengths of both humans and AI may succeed. In my PhD I am extending the current capabilities of human-machine collaboration to achieve complex tasks through conversation. In particular, AI partnerships to generate and refine program code.

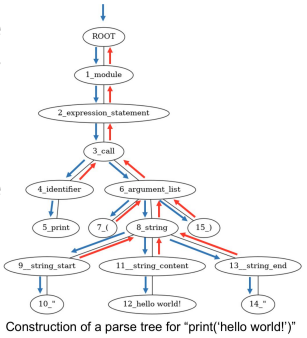
Why Code?

Complex tasks are abundant. However, programming benefits from being fully digitized while still involving all essential aspects of complex problem solving. Its structured and brittle nature poses an additional challenge since we generally communicate through diffuse ideas rather than concrete implementations. The wealth of data available through GitHub and the open source community provides the ideal environment for data hungry deep learning. Given the ubiquity of software, any improvements can deliver substantial benefits.

A PhD is best defined by the questions it will answer

1. What are the unique challenges of code generation and how can neural architectures synthesise correct code optimally?

While a programming language (PL) structure is tightly enforced, humans generate code one word at a time. Current neural language models follow a similar auto-regressive pattern risking malformed output. To solve this, I generate directly the tree representation of code where the PLs grammar can be enforced, guaranteeing correctness. However, I am exploring the use of grounding for auto-regressive generation with error messages, improving semantic awareness and simplifying the decoding process, putting into question the need for state-of-the-art tree decoders.



Construction of a parse tree for "print('hello world!')"

2. How can neural architectures track complex goals in a conversational setting?

Current language models see a conversation as a flat paragraph of text, reading all text contained in one. However, as a new word is added the process repeats, resulting in the inefficient n^2 sequence length complexity that plagues current Transformer models. In contrast, we as humans highlight important elements as they arise in a conversation, like a concept bubble in a graph, and throw away the rest. Dense Graph Memory is this idea framed for neural models, using dense keys from graph attention. However, there are challenges to overcome for neural memory, such as the weakening signal traveling through time in an LSTM.

3. How can relevant external information be leveraged in neural architectures to improve prediction?

A model's capacity is the amount of information it can store for solving a task. Models like GPT-3 have shown huge capacity for memorization yet no way to integrate new knowledge without re-training. This ability to integrate new external knowledge is crucial for complex tasks like programming where documentation changes rapidly. I published my first step in dynamically increasing model capacity in Relevance Transformer at SIGIR 2020, where pseudo-relevance feedback was used in a novel way to improve code generation by biasing a Transformer's output. I am currently pursuing the integration of external knowledge similarly to question 2.

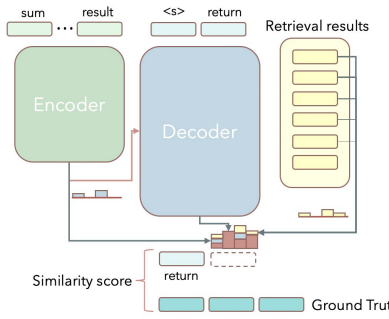


Diagram of the Relevance Transformer architecture