Cheat Codes to Quantify Missing Source Information in Neural Machine Translation

Edinburgh University of Edinburgh Natural Language Processing NLP

THE UNIVERSITY of EDINBURGH informatics

Proyag Pal **Kenneth Heafield**

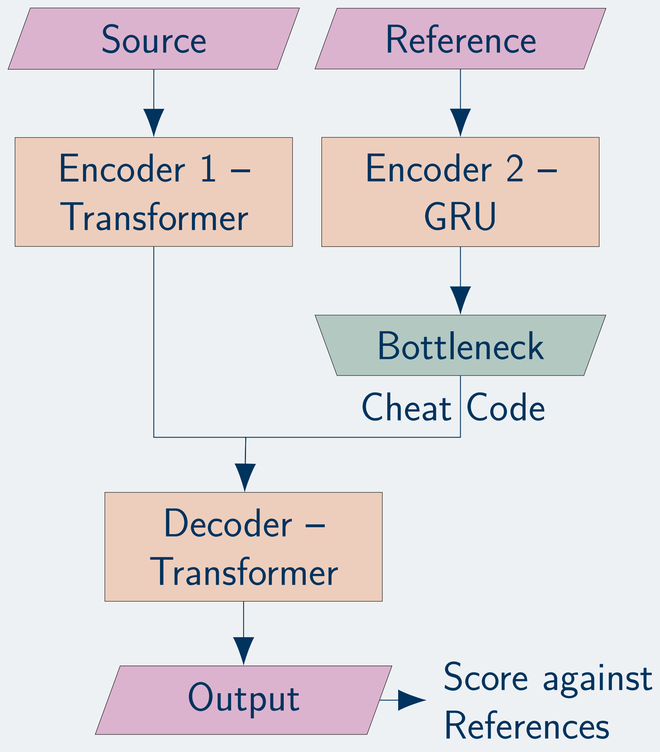
Overview: Why Cheat?

How much information $\mathbf{H}(\mathbf{t}|\mathbf{s})$ is in the target sentence **t** that is not present in the source **s**?
Method: give the decoder a representation of the answer (a **cheat code**) as an additional input, but bottleneck it.
How small a cheat code is useful?
How big a cheat code reproduces the target?

How Much Cheating?

We can vary the cheat code size and observe its effect on test set scores.

How to Cheat: Dual-Encoder



Interpolating Cheat Codes

Interpolate between reference encodings, use as the cheat code:
$\lambda \cdot \text{enc}(\text{refA}) + (1 - \lambda) \cdot \text{enc}(\text{refB})$

Conclusions

Even a single float conveys useful information about the target.
Model almost reproduces the target with 2 floats per token.
Continuous space of cheat codes between references exists.

More details?

Paper: proyag.github.io/files/papers/cheat-codes.pdf
Code: github.com/Proyag/marian-dev/tree/cheat-codes
Contact: proyag.pal@ed.ac.uk

This work was supported by funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 825303 (Bergamot).