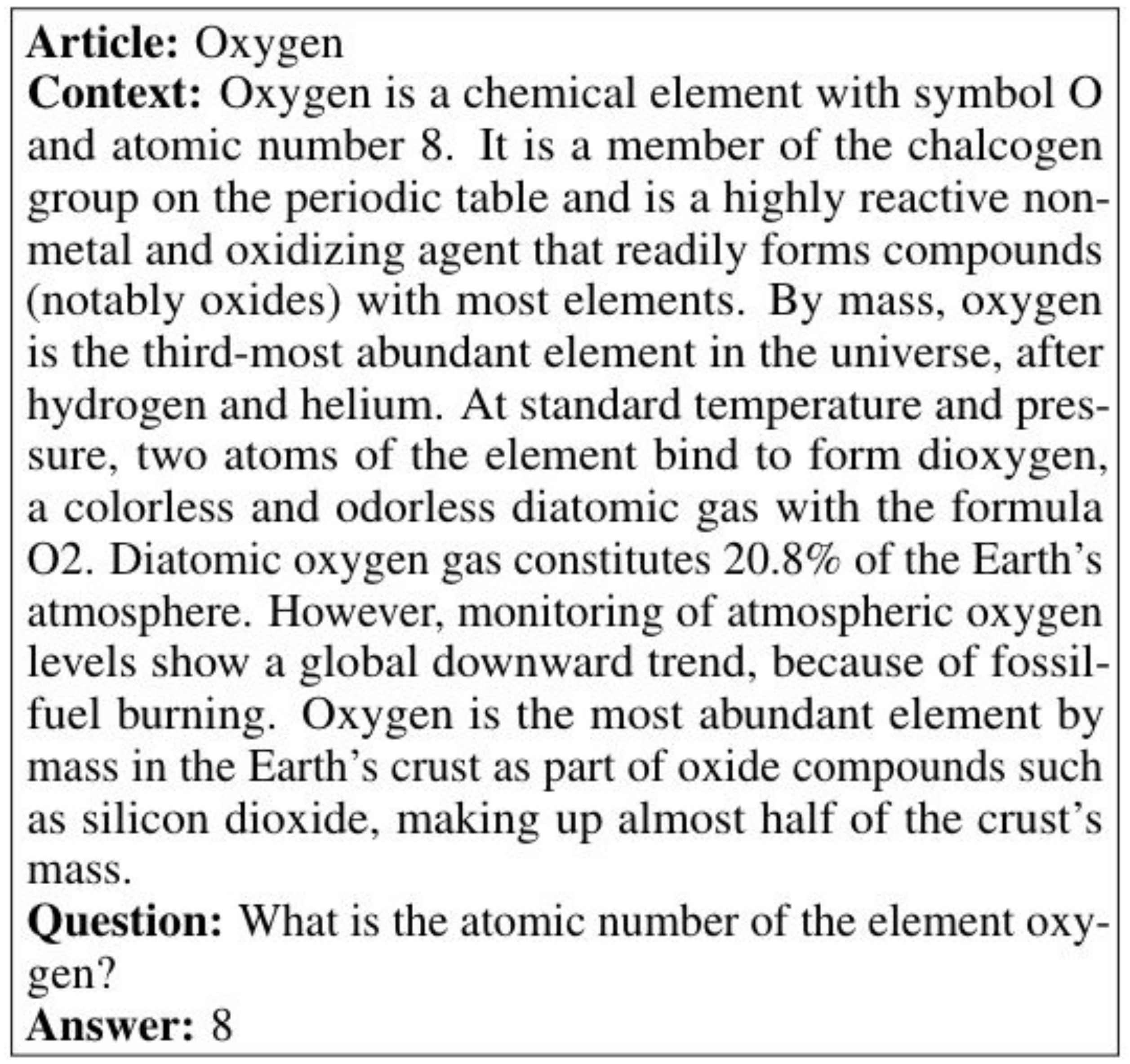
Adversarially Improving Adversarial Performance of QA models

Yoni Lerner
Department of Computer Science

Problem

SQuAD question answering (QA) tasks have been a popular operationalisation of machine comprehension. Recent results show state-of-the-art QA models are highly susceptible to very simple adversarial examples that humans ignore, suggesting they may not be achieving machine comprehension. In this project I created a model to generate adversarial examples usable as data augmentation to train more robust QA models.

Data & Task



Article: Oxygen
Context: Oxygen is a chemical element with symbol O and atomic number 8. It is a member of the chalcogen group on the periodic table and is a highly reactive nonmetal and oxidizing agent that readily forms compounds (notably oxides) with most elements. By mass, oxygen is the third-most abundant element in the universe, after hydrogen and helium. At standard temperature and pressure, two atoms of the element bind to form dioxygen, a colorless and odorless diatomic gas with the formula O2. Diatomic oxygen gas constitutes 20.8% of the Earth's atmosphere. However, monitoring of atmospheric oxygen levels show a global downward trend, because of fossil-fuel burning. Oxygen is the most abundant element by mass in the Earth's crust as part of oxide compounds such as silicon dioxide, making up almost half of the crust's mass.
Question: What is the atomic number of the element oxygen?
Answer: 8

The data used was the SQuAD 2.0 data consisting of (context, question, answer) triplets. The task for QA models was to predict the answer as a substring of the context given the context and query. The task for the project was to adversarially augment the context.

Approach

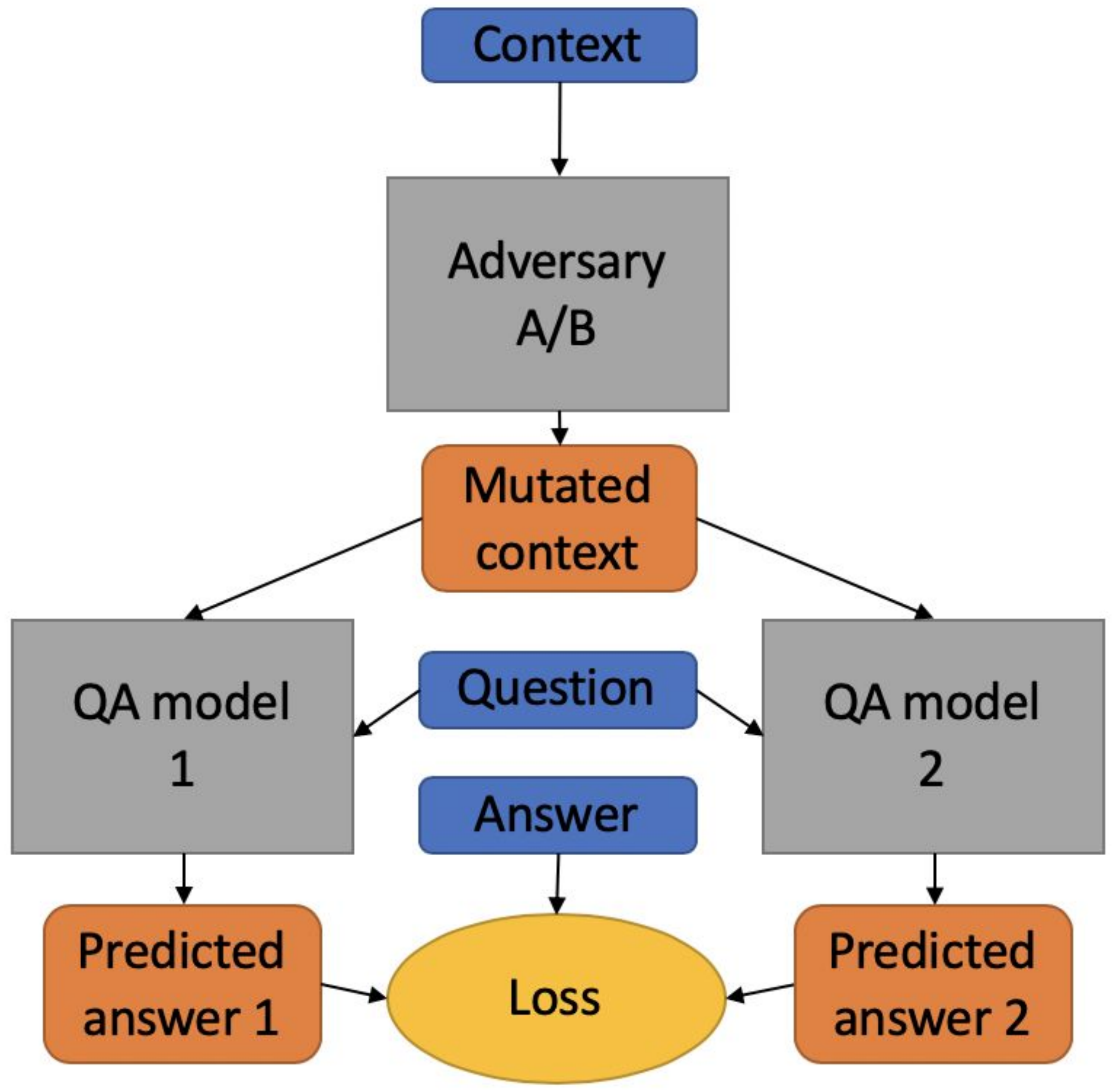
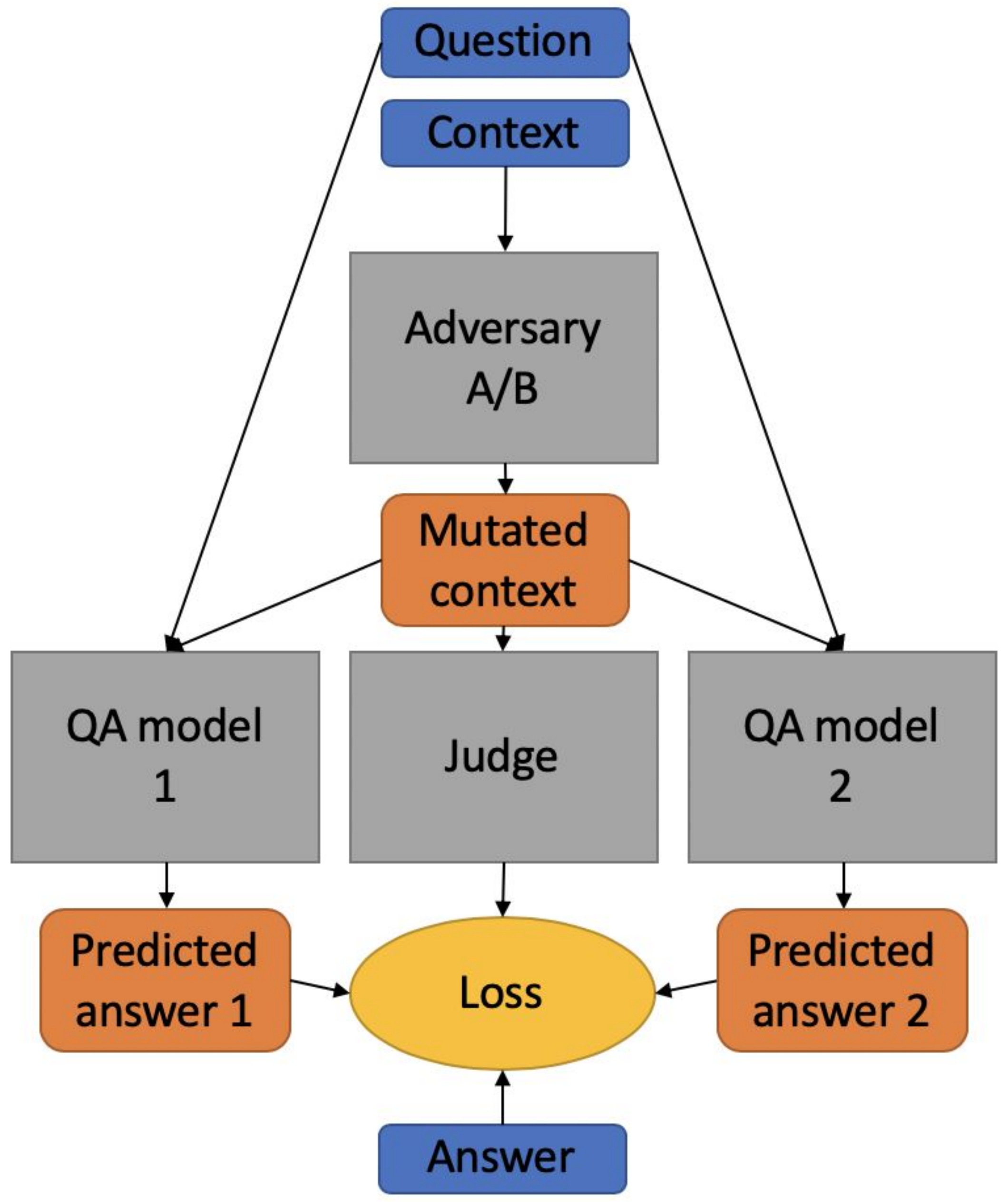


Fig 1. (Top) The Tag Team Adversary architecture with fixed QA models.
Fig 2. (Bottom) The Tag Team Adversary architecture with live QA models.



Results and Analysis

Unfortunately, TTA proved too complex to stably implement for the scope of this project. To evaluate the success of TTA, we would use several approaches. First, using the fixed-QA version of TTA, an augmented dataset would be created. Its usability as training data would be tested by training a standard SQuAD QA model on the augmented data, then testing its performance on SQuAD, AddSent-augmented SQuAD, and AddAny-augmented SQuAD. Then, a qualitative examination of a random sample of the dataset would check the validity of generated contexts and the diversity and quality of mutations. For the non-QA-fixed version, the QA model would be trained and its performance would be evaluated on SQuAD, AddSent-augmented SQuAD, and AddAny-augmented SQuAD.

Conclusions & Future Work

TTA is an adversarial architecture for adversarial example generation in SQuAD. By relying on a tag-team competition between two pairs of networks (and potentially a judge network), it maintains validity of context while incentivizing creation of difficult mutations. Future work would focus on simplifying this architecture, as one of its main limitations is its high complexity and parameter-count and resulting instability. For example, a better adversary architecture could help create specific desired types of mutations with far less parameters (for example one that edited the context instead of reproducing a whole new one sequence-to-sequence).

References

[1] Ian Goodfellow et al. “Generative Adversarial Nets”. In: Advances in Neural InformationProcessing Systems 27. Ed. by Z. Ghahramani et al. Curran Associates, Inc., 2014, pp. 2672–2680. URL: http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf.

[2] Robin Jia and Percy Liang. “Adversarial Examples for Evaluating Reading ComprehensionSystems”. In: CoRRabs/1707.07328 (2017). arXiv: 1707.07328. URL: http://arxiv.org/abs/1707.07328.

[3] Pranav Rajpurkar et al. “SQuAD: 100, 000+ Questions for Machine Comprehension of Text”. In: CoRRabs/1606.05250 (2016). arXiv: 1606.05250. URL: http://arxiv.org/abs/1606.05250.