Data Augmentation with Adversarial Examples and Back-Translation

Stanford CS224N Default Project (Robust QA)

TA Mentor: Fenglu Hong

Yuzu Ido
Department of Computer Science
Stanford University
yuzu@stanford.edu

Stephan Sharkov
Department of Computer Science
Stanford University
stpshrkv@stanford.edu

Eunice Yang
Department of Computer Science
Stanford University
eunicey@stanford.edu

Abstract

Though question answering (QA) systems serve essential roles online and drive the websites of many Internet-based companies, they often perform poorly because of NLP models' inability to learn deeply beyond their given training distribution. Here, we seek to improve upon basic QA models by implementing a robust QA system that can generalize to unseen, out-of-domain data. Our QA model is based on data augmentation, primarily with adversarial examples generated by a language model replacing one masked token in our original data, and backtranslation using Russian as a pivot language. Compared to the DistilBERT baseline dev F1 and EM of 47.72 and 30.63 respectively, our best data augmentation-based model achieved dev F1 and EM of 49.07 and 31.68, and test F1 and EM of 59.48 and 40.51.

1 Introduction

Question answering (QA) remains a critical natural language processing (NLP) task and long-standing challenge in the field of AI. On a fundamental level, QA systems take in a question and related paragraph as input, then attempt to output the correct answer to the given question. While QA may appear relatively straightforward, this type of reading comprehension problem requires models to have a high level of natural language understanding. It is very challenging to create models that can deeply comprehend text, learn relationships between words and phrases in the text, *and* generalize these learnings to new domains.

There has been considerable progress over the last few years with the development of transformer-based pretrained models such as BERT [1] and ALBERT [2]. These models have made significant breakthroughs in language representation learning with greatly improved performance and higher efficiency. However, generalizability of even these state-of-the-art models beyond the training distribution remains a significant weakness. Models may perform well on test sets filled only with previously seen domains, but researchers have found that this can often be due to the learning of superficial correlations [3] or falliable heuristics [4] which fail to apply to unseen domains. In the real world, this is a critical flaw as QA systems may rarely receive strictly "in-domain" test examples and will subsequently suffer in accuracy and performance.

In this project, we sought to develop a QA system that is more robust to out-of-distribution data as compared to current models. We ultimately chose to use data augmentation as our core technique to improve generalizability, given that it has consistently been used to reduce overall overfitting on

brittle correlations and improve the performance of other machine learning models. At its core, this approach seeks to increase training data and in particular the diversity of the examples—without actually having to collect any new data. Our goal was to encode additional examples with label preserving invariances to increase the diversity of our training set. More specifically, we utilized BAE (BERT-based Adversarial Examples) and backtranslation to augment our datasets, then analyzed how these approaches improved upon our baseline QA model's performance.

2 Related Work

A number of research groups have thoroughly assessed the generalizability of natural language models and found critical weaknesses in many of these systems. An earlier paper by Jia et al. published in 2017 ran an adversarial evaluation on 16 published models using SQuAD and found that the average F1 score dropped by more than half from 75% to 36% [5]. Another study by Gururrangan et al. found that the premise/hypothesis structure of natural language inference dataset actually results in annotation artifacts [6]. Linguistic phenomena (i.e. negation and vagueness) are correlated with specific inference classes and likely result in the overestimation of the degree to which natural language inference (NLI) models are actually comprehending the texts they are given. Note that both of these findings were prior to landmark developments like the BERT model.

However, recent studies have found similar issues with robustness in BERT and other advanced models. After identifying three common syntactic heuristics in NLI models, McCoy et al. developed a HANS evaluation set (Heuristic Analysis for NLI Systems) containing examples where these heuristics tend to fail [4]. All of the evaluated models, including BERT, performed very poorly on HANS, exhibiting logic that is aligned with these heuristics but do not demonstrate correct inference. They did note though that BERT outperformed other models on two of the three heuristic types. In another study, McCoy et al. trained 100 instances of BERT on a MNLI dataset and again evaluated on HANS, which served as a measure of syntactic generalization [7]. Though all these instances performed similarly as expected on MNLI's dev set with accuracies around 84%, there was significant variation in performance on HANS with accuracy ranging from 0% to 66.2%. These issues with generalizability extend to applications outside of QA, with one recent study by Khambete et al. finding that BERT performance on the new electronic health record (EHR) corpus was higher when train and test content were more similar [8]. Additionally, test performance only improved when BERT was trained on EHR from more medical specialties.

Data augmentation presents a promising but relatively unexplored approach to improve the generalizability of NLP models. Feng at al. classifies data augmentation approaches using three main categories: rule-based techniques, example interpolation techniques, and model-based techniques [9]. Rule-based techniques tend to be more primitive with easy-to-compute transformations, such as the Wei et al. Easy Data Augmentation approach involving token-level operations like "deletion" and "swap" [10]. Exam interpolation techniques involve mixing input examples and labels. We focus in particular on the third model-based set of techniques that leverage other NLP models to generate new data. One recent example of this includes Riberio et al. Semantically Equivalent Adversaries, which found that training on adversarial augmented data maintains accuracy and drop sensitivity to bugs by 10% [11]. Another interesting exam interpolation technique from Bornea et al. created a multilingual dataset 14 times the initial English dataset size [12]. This decreased reliance on unnecessary dependencies of language specifics and increased generalizability across languages in multilingual QA tasks.

Our project was inspired by two model-based techniques to perform data augmentation for our robust QA task. The first was Garg and Ramkrishnan's BAE (BERT-based Adversarial Examples) approach [13]. BAE was developed to attack modern text classification models, generating adversarial examples in two ways: replacing a token within a sentence with a new, BERT-predicted token, and inserting a new word token in the original sentence. To avoid issues with sentence grammar and context, the top K tokens with high semantic similarity to the original text are used. BAE then picks a token candidate that either most decreases successful prediction probability or, if there is significant misclassification, is most similar to the original. This process is repeated for every word in the sentence in order of word importance until the algorithm either misclassifies the sentence (attack success) or runs out of words to perturb (attack fail). Bae achieved a 40-80 percent drop in models' test accuracy while achieving high human evaluation scores, almost two times more effective than previous adversarial examples.

We were also interested in backtranslation approaches and built upon the basic method by Sennrich et al [14]. Here, the researchers sought to avoid changing their neural network architecture and just train with monolingual data for a neural machine translation task. Sentences were translated from a German monolingual dataset to English then "back-translated" to German. Training on these additional examples resulted in significant improvements on the baseline models of +2.8-3.7 BLEU, and additional tests on a lower-resourced task (Turkish/English) also showed interesting, substantial improvements of +2.1-3.4 BLEU.

3 Approach: Data Augmentation

3.1 Baseline

The baseline is a DistilBERT transformer model finetuned on all the original training data [15], as described in the RobustQA handout. DistilBERT is a compressed version of the original BERT model, with size reduced by 40% but retainment of 97% of language understanding capabilities and 60% increase in speed. The general architecture remains largely the same, with a 2x reduction in layers.

3.2 BAE: BERT-based Adversarial Examples

Our first approach to data augmentation is a variation on Garg and Ramakrishnan's BAE [13]. In our implementation, we first pick n random indices $(i_1, \ldots, i_n)$ to mask a token, from a uniform distribution over all positions in the question after the [CLS] token and before the "?" token, with replacement for coding convenience.

Note that this is different from the original approach in BAE, which lists tokens by word importance then iteratively replaces them, one by one, from the most important one. This is because the BAE authors were examining a sentiment classification task, for which overall sentence similarity is sufficient, even with a few synonyms; however, for a question answering task, we expected the most important words to often be necessary for answering correctly, since they have high factual content. Randomization seemed like a good approach instead of selecting high importance words which would damage the question content, or low importance words, which would not affect the model enough.

Then, after selecting n random indices, for each random index i, we do the following procedure 1.

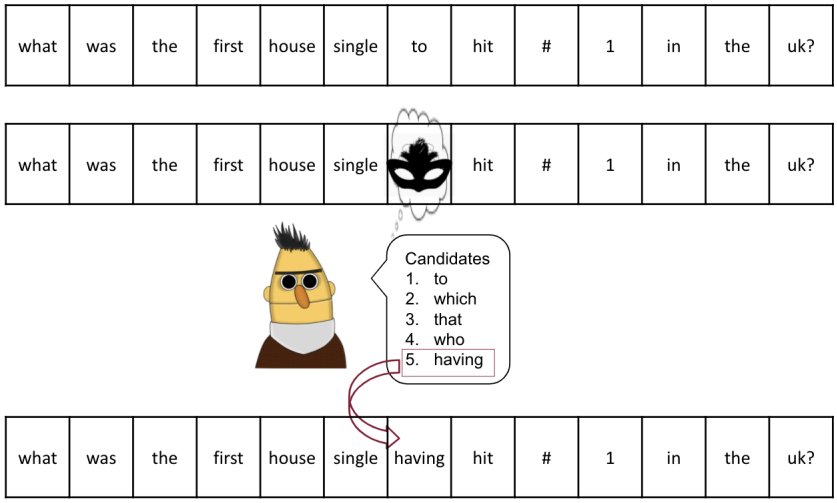

Figure 1: BAE Schematic.

First, we input the entire example (context, question, answer) and ask DistilBERT to predict the top $k = 5$ candidates to fill in the mask at i in the question. Then, we filter for words that keep the question relatively similar to the original in meaning according to a similarity score of above 0.9 based on the Universal Sentence Encoder, for which we use an existing implementation [17].

Why is this necessary? The BAE authors provide the following example: given the original sentence "The food is good," if we mask "good," BERT may predict "bad." The sentence "The food is bad" is grammatically and contextually correct, but alters the meaning of our sentence.

Finally, among the filtered candidates, we want to choose the candidate that is the most adversarial (that is, the QA model is likely to predict wrong); we do this by choosing the candidate with the lowest similarity score that passes our 0.9 cutoff. We replace the masked token with the chosen BERT prediction, then repeat with the next random index, using the newly modified question as our input.

3.3 Back-translation

Our other approach to data augmentation is back-translation: we translate our given datasets' questions from English to Russian using the Helsinki-NLP OPUS-MT-EN-RU model, and back to English with the OPUS-MT-RU-EN model [16]. The Helsinki-NLP models were chosen due to high similarity of data formatting to our given datasets, and thus ease of integration. Russian was chosen as the pivot language because we had a Russian speaker on our team to evaluate the intermediate translations, and also because English-Russian machine translation is well developed enough to give reasonable results, while still introducing differences after back translation.

4 Experiments

4.1 Data

We are working on the RobustQA default project, which has a question answering task. We have been provided with three "in-domain" datasets: SQuAD, NewsQA, and Natural Questions. Each of the three datasets contains 50,000 training elements for a $\mathbf{indomain\_train}$ set, and contain 10507, 4212, and 12836 validation elements respectively for a $\mathbf{indomain\_val}$ set. The datasets have been taken from Wikipedia and news articles.

We have also been provided with three "out-of-domain" datasets: DuoRC, RACE, and Relation Extraction. These datasets contain 1248, 419, and 2693 elements respectively for testing, comprising the held out test set $\mathbf{oodomain\_test}$. From each of these three sets, approximately 127 elements have been provided to us for our training and validation. These oo-domain datasets have been taken from movie reviews, exams, and Wikipedia.

Example from SQuAD 2.0

Context: Southern California, often abbreviated SoCal, is a geographic and cultural region that generally comprises California's southernmost 10 counties. The region is traditionally described as "eight counties", based on demographics and economic ties: Imperial, Los Angeles, Orange, Riverside, San Bernardino, San Diego, Santa Barbara, and Ventura. The more extensive 10-county definition, including Kern and San Luis Obispo counties, is also used based on historical political divisions. Southern California is a major economic center for the state of California and the United States.
Q: What is Southern California often abbreviated as?
A: SoCal

4.2 Evaluation method

Our main quantitative evaluation metrics is F1, and our secondary quantitative metric is EM, as described in the RobustQA handout. As a qualitative metric, we visually inspected the DistilBERT predictions to make sure our BAE adversarial examples seemed like reasonable English sentences. Similarly, for back-translation, our Russian team member examined the intermediate English-to-Russian machine translations.

4.3 Experimental details

We first created BAE examples on the SQuAD dataset with $n = 1$ and $n = 3$ tokens masked and replaced. Since $n = 1$ had higher $F1$ when we finetuned the baseline on just the augmented SQuAD examples (which took 0.5 hours), we used $n = 1$ hyperparameter for the rest of the BAE examples.

We created BAE and back-translated datasets from all provided datasets, the three in-domain (SQuAD, NewsQA, and Natural Questions) and out-of-domain (RACE, Relation Extraction, DuoRC).

We first loaded the parameters from the baseline, then chose various combinations of our adversarial examples to finetune the model on. Initially, we used all the default hyperparameters from the baseline to choose our best combination of adversarial examples. All finetunings containing both BAE and backtranslation for all three in-domain datasets took about 4 hours. The time for finetuning on just the out-of-domain augmented datasets, if separated from the in-domain datasets, took under 5 minutes.

Once we decided on the best model, which was first finetuning on the in-domain augmented examples (BAE and backtranslation randomized) and later the out-of-domain augmented examples (BAE and backtranslation randomized) with 49.07 F1 score, we started hyperparameter tuning. Since the baseline default was $\alpha = 3 \times 10^{-5}$, we tested new learning rates of $\alpha = 3 \times 10^{-4}$ and 3×10^{-6}, inspired by a log scale. Then, that did not improve results on the dev set, so we chose $\alpha = 3 \times 10^{-5}$, and we tuned the number of epochs. The default in the baseline was 3, so we tested 1 and 2 since we had an assumption that we are overfitting on our training data. Epochs 1 and 2 got the same F-1 as 3, but improved with EM, but we decided to proceed with 2 epochs to have sufficient training time. Then, we tuned the batch size. The default is 16, so we tested 8 and 32, out of which we still chose 16 batch size, since new batch sizes resulted in lower F-1 scores.

4.4 Results

We provided the training loss graphs of our best model below:

Figure 2: Best model's loss during training on in-domain BAE and BT examples

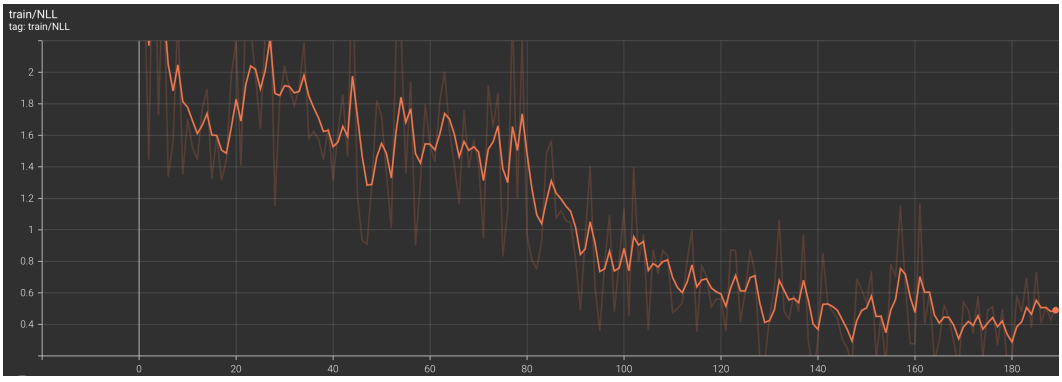

Figure 3: Best model's loss during training on out-of-domain BAE and BT examples

Our validation evaluation metrics for a representative selection of our models are reported in 1, as well as our test evaluation metrics in 2 for two of our best models [1]

F1 Rank	Model	F1	EM	α	#Ep	bS
-	Baseline DistilBERT	47.72	30.63	3e-5	3	16
1	IN-BAE + IN-BT < OUT-BAE + OUT-BT	49.07	31.68	3e-5	1	16
1	IN-BAE + IN-BT < OUT-BAE + OUT-BT	49.07	31.68	3e-5	2	16
1	IN-BAE + IN-BT < OUT-BAE + OUT-BT	49.07	31.58	3e-5	3	16
2	IN-BAE < IN-BT < OUT-BAE + OUT-BT	48.76	31.68	3e-5	3	16
3	IN-BAE + IN-BT < OUT-BAE + OUT-BT	48.63	31.15	3e-5	2	32
4	IN-BAE + IN-BT < OUT-BAE + OUT-BT	48.59	31.15	3e-5	2	8
5	IN-BAE < OUT-BAE	48.48	31.15	3e-5	3	16
6	IN-BT	48.26	30.63	3e-5	3	16
7	IN-BAE + IN-BT + OUT-BAE + OUT-BT	48.2	30.1	3e-5	3	16
8	IN-BAE + IN-BT < OUT-BAE + OUT-BT	48.06	32.2	3e-6	3	16
9	IN-BT < OUT-BT	48.05	30.63	3e-5	3	16
10	IN-BAE	47.99	30.89	3e-5	3	16
11	IN-BAE + IN-BT < OUT-BAE + OUT-BT	44.47	27.75	3e-4	3	16

Table 1: Evaluation Metrics on Dev Set

F1 Rank	Model	F1	EM	α	#Ep	bS
1	IN-BAE + IN-BT < OUT-BAE + OUT-BT	59.48	40.51	3e-5	3	16
1	IN-BAE + IN-BT < OUT-BAE + OUT-BT	59.48	40.51	3e-5	2	16
1	IN-BAE + IN-BT < OUT-BAE + OUT-BT	59.48	40.51	3e-5	1	16

Table 2: Evaluation Metrics on Test Set

Our assumption that creating and adding OUT-BAE and OUT-BT would increase model performance turned out to be true, as that increased our F-1 scores for all models; we can see 5 performed better than 10, and most models performed better than 6 or 10. We also expected mixing IN-BAE and IN-BT with each other to be better than layering them, since the latter might teach the model brittle correlations of the first set that it would need to unlearn with the second; this was supported as 1 performed better than 2.

We were surprised by how high our test metrics were relative to our dev metrics, but this seemed to be the trend for other groups on the leaderboard. We expected a mix of all data in 7 to result in better F1 score than layering; however, 2 and 1 were better. We additionally expected including pure OUT to improve our score, but that only decreased it below the baseline, which is an interesting observation.

5 Analysis

Failed Prediction 1

Context: Nikola Tesla (Serbian Cyrillic: <We could not paste Cyrillic>; 10 July 1856 - 7 January 1943) was a Serbian American inventor, electric engineer, mechanical engineer, physicist, and futurist best known for his contributions to the design of the modern alternating current (AC) electrical supply system.
Q: In what year did Tesla die?
A: 1943
Prediction: 10 July 1856 - 7 January 1943

As in this above example, we saw that the model often retrieved passages that contained, but were longer than, the gold label answer. One possible reason is that more of the in-domain examples seem to have a longer answer relative to the out-of-domain ones. Thus, the machine may have learned this irrelevant feature specific to the in-domain distribution, which we did not target in performing BAE

[1] Abbreviations: < - layered on top; + - joint and randomised; IN - all in domain (SQuAD, Nat, NewsQA); OUT - out of domain (DuoRC, RACE, RE); BAE - BERT Adversarial Training; BT - Back Translation, #Ep - numEpochs, bS - batchSize.

or backtranslation of only the questions. Another potential difficulty with this particular example is the inclusion of some Cyrillic in addition to the English text.

Failed Prediction 2

Context: (CNN) – Actor Gary Coleman is in critical condition in a Provo, Utah, hospital, a hospital spokeswoman said Thursday. Janet Frank, the spokeswoman for Utah Valley Regional Medical Center, confirmed that Coleman, 42, was being treated there after being admitted on Wednesday. Frank would not release any other information. Calls to Coleman's publicist were not immediately returned. . . . CNN's Brittany Kaplan contributed to this report.
Q: What is the name of the hospital where Gary Coleman was admitted?
A: Utah Valley Regional Medical Center
Prediction: Provo, Utah,

The failed prediction suggests that the model is weak to punctuation such as the comma at the start and end of "Utah" in the passage. While it is possible to select punctuation as the random token to mask and replace using BAE, there are typically fewer candidates available to substitute for a punctuation mark than a normal word. This suggests that the generated examples for punctuation are not widely as spread across the embedding space as words, causing them to remain difficult for the model.

6 Conclusion

In our project we were able to implement our versions of BAE data augmentation and backtranslation, which proved to improve our metric scores when added to the baseline model. We learned to use transformer-based models, especially BERT, and adapt them for different models, perform hyperparameter tuning and different ways of mixing or layering data, and examine which ones help to get better results.

Only changing one word for BAE resulted in examples which were quite similar to the originals, meaning they were not as adversarial as possible. Certain model configurations thus suggested overfitting. At the same time, changing too many words (we tried 3) resulted in lower metric scores.

For future work, it is promising to tackle the challenge of accidentally masking highly factual content in BAE. As well as consider implementing BAE's approach on removing words that contribute the most to correctly predicting an answer. In addition, exploring either BAE or back translation on the whole example, instead of just the question, would provide more changes during augmentation and thus examples that are farther from the in domain distribution; we were unable to do this due to limitations in computational power and time.

References

[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. 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 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

[2] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942.*

[3] Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. *arXiv preprint arXiv:2004.10964.*

[4] R Thomas McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syn- tactic heuristics in natural language inference. In *Association for Computational Linguistics (ACL).*

[5] R. Jia and P. Liang, "Adversarial examples for evaluating reading comprehension systems," Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, 2017.

[6] S. Gururangan, S. Swayamdipta, O. Levy, R. Schwartz, S. Bowman, and N. A. Smith, "Annotation artifacts in Natural Language Inference Data," Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers), 2018.

[7] R. T. McCoy, J. Min, and T. Linzen, "Berts of a feather do not generalize together: Large variability in generalization across models with similar test set performance," Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP, 2020.

[8] M. Khambete, W. Su, J. Garcia, M. Badgeley. 2021. Quantification of BERT Diagnosis Generalizability Across Medical Specialties Using Semantic Dataset Distance. *arXiv preprint arxiv.org/abs/2008.06606.*

[9] S. Feng, V. Gangal, J. Wei, S. Chandar, S. Vosoughi, T. Mitamura, and E. Hovy, "A survey of Data Augmentation Approaches for NLP," Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, 2021.

[10] J. Wei and K. Zou, "Eda: Easy data augmentation techniques for boosting performance on text classification tasks," Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), 2019.

[11] M. T. Ribeiro, S. Singh, and C. Guestrin, "Semantically equivalent adversarial rules for debugging NLP models," Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2018.

[12] M. Bornea, L. Pan, S. Rosenthal, R. Florian, A. Sil. 2020. Multilingual Transfer Learning for QA Using Translation as Data Augmentation. *arXiv preprint arxiv.org/abs/2012.05958.*

[13] S. Garg and G. Ramakrishnan, "Bae: Bert-based adversarial examples for text classification," Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2020.

[14] R. Sennrich, B. Haddow, and A. Birch, "Improving neural machine translation models with monolingual data," Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2016.

[15] S. Victor, D. Lysandre, C. Julien, W. Thomas. 2020. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv preprint arxiv.org/pdf/1910.01108.*

[16] J. Tiedemann, S. Thottingal. 2020. "OPUS-MT — Building open translation services for the World," Proceedings of the 22nd Annual Conferenec of the European Association for Machine Translation (EAMT), 2020.

[17] D. Cer, Y. Yang, S.-yi Kong, N. Hua, N. Limtiaco, R. St. John, N. Constant, M. Guajardo-Cespedes, S. Yuan, C. Tar, B. Strope, and R. Kurzweil, "Universal sentence encoder for English," Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, 2018.

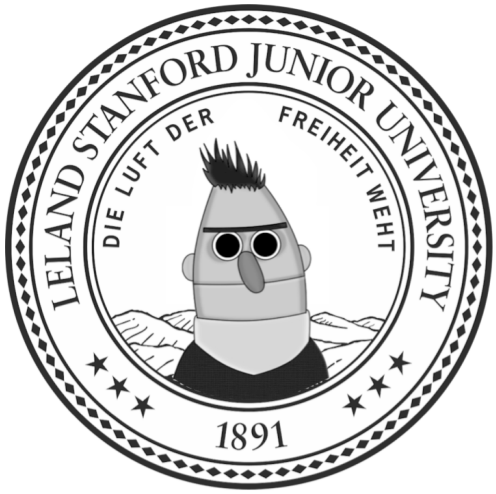

Figure 4: **Leland Bert Stanford Junior University Logo.** Leland Bert Stanford is a yellow Muppet character on beloved PBS show Sesame Street known for singing the hit song, "Doin' the Pigeon." In honor of Bert's incredible impact on all of humanity, a university and natural language model were named after him.

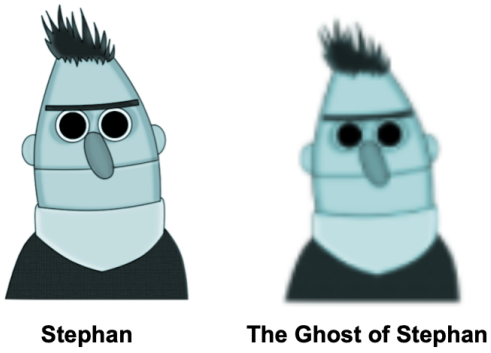

Figure 5: **4K HD Photograph taken of Stephan during Dead Week.** The Ghost of Stephan was responsible for typing in Terminal when Stephan was nowhere near the keyboard and hemorrhaging our Azure credits. Unfortunately, the Ghost of Stephan was not credited as an author for this paper and we are currently embroiled in a serious authorship dispute.