NOLEdge: Using AI to Develop an Intelligent Search Tool for a University Department's Web Domain

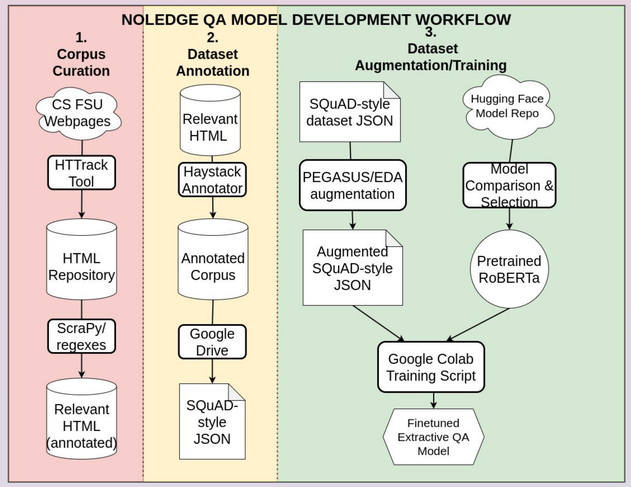
Presenter: Colin MacRae; Honors in the Major Committee Members: Dr. Sonia Haiduc, Dr. Shayok Chakraborty, & Dr. Adrian Barbu
Department of Computer Science, Florida State University

Abstract
This project involves using state of the art artificial intelligence models trained for processing text to create an intelligent search tool for the Florida State University Computer Science (FSU CS) department's web domain, named NOLEdge. A dataset created from text scraped from the FSU CS web domain was fed to the model to improve NOLEdge's performance. The use of data augmentation to improve the model's performance was also explored. The best-performing version of the fine-tuned model was packaged into a Chrome Extension for convenient use by FSU CS students and staff.

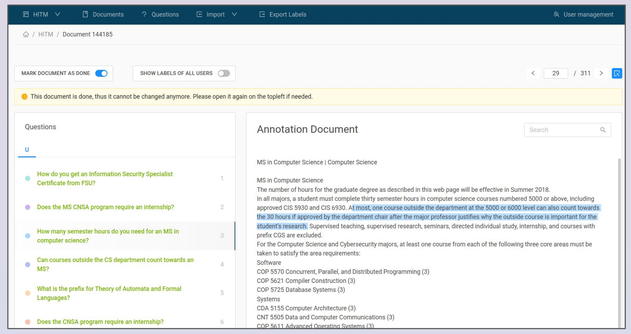
Motivation
- Keyword searches only successful if desired phrase contains keyword
- Incorrect keyword search leads to no relevant results and/or irrelevant results
- AI improves the search process by
  - allowing user to input a full question (ex: "When are grad apps due?")
  - matching semantics instead of keywords

Research Questions:
1. Of BERT, RoBERTa, ALBERT, and DistilBERT, which flavor of BERT is optimal for deployment as an extractive QA model for an intelligent search tool?
2. How will fine-tuning affect the chosen model's performance?
3. How do different types of textual data augmentation on the training set affect the fine-tuned model's performance?

Development Phases:
1. Create fine-tuning dataset
2. Select best pretrained model
3. Augment the dataset using various methods
4. Fine-tune selected pretrained model on augmented/unaugmented datasets
5. Determine best fine-tuned model
6. Deploy fine-tuned model

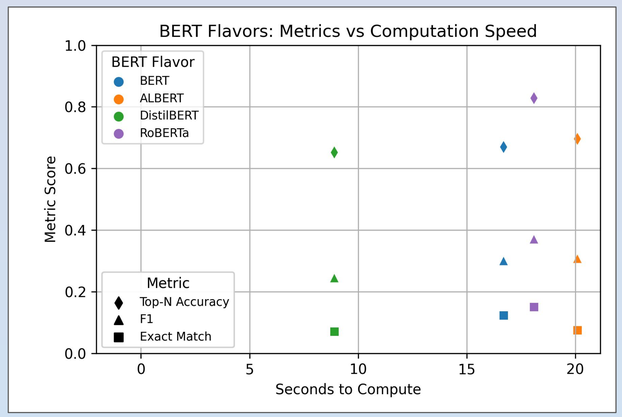


The three phases of **NOLEdge model development**



Screenshot of **Deepset AI Haystack Annotation Tool** with one of the documents scraped from an FSU CS webpage

Selecting A Base Model
- base model needed to be accurate without being too slow
- models often sacrifice accuracy for speed
- RoBERTa flavor performed best although ALBERT flavor was larger



Comparison of various **BERT flavor metrics** on a holdout set

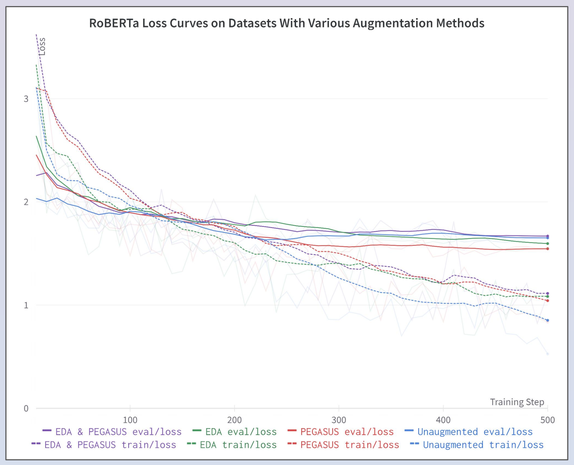
Augmenting the Dataset
Two forms of textual data augmentation were used:
1) **Symbolic augmentation:** involves actions like word replacement, word shuffling, or word deletion.
2) **Neural augmentation:** involves the use of other AI models to create augmented data mimicking aspects of the original such as style or content.

```
ORIGINAL: Where did the award come from?
PEGASUS AUG: Where did the award come from?
PEGASUS AUG: What happened to the award?
PEGASUS AUG: Where did the prize come from?
PEGASUS AUG: The award came from where?
```

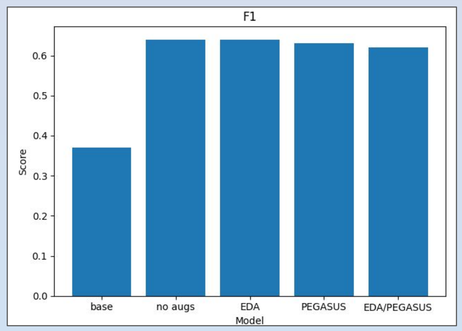
```
ORIGINAL: Where did the award come from?
EDA AUG: where did the present come from
EDA AUG: did the award come from
EDA AUG: where did the award awarding come from
EDA AUG: did the award come from
```

Examples of both symbolic (EDA) and neural (PEGASUS) **textual data augmentation.** Notice that neural augmentation does a better job of retaining the *meaning* of the original phrase.

Fine-tuning Model on the Datasets
The pretrained RoBERTa model was fine-tuned on the datasets and loss curves were recorder using WandB to gain insight on the model's fitting capacities.

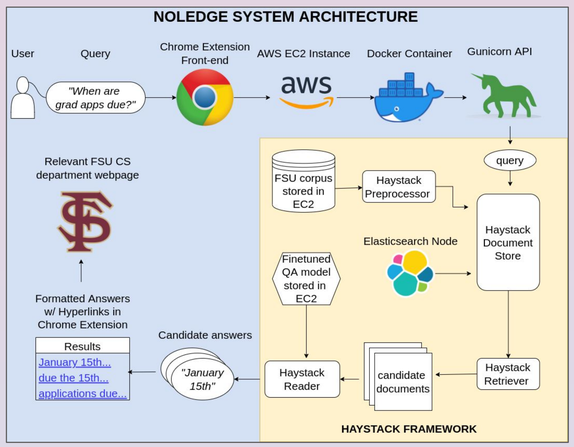


Evaluating the Fine-tuned Models

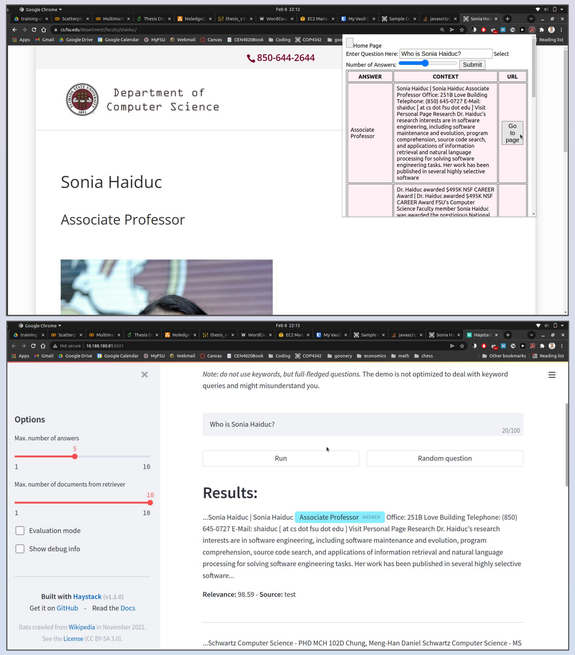


Comparison of various **fine-tuned model F1 scores** on a holdout set

Deploying the Chosen Fine-tuned Model
- selected the model trained with dataset augmented using PEGASUS
- deployed model via a backend using AWS and Python and a frontend Chrome Extension



The **architecture** of the NOLEdge system and flow of information



Screenshot of the NOLEdge **front-ends** in action (Chrome Extension above, Haystack Streamlit Docker image below)

References
[1]V. Sanh, et.al, "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter," *arXiv preprint arXiv:1910.01108*, 2019.
[2]Y. Liu *et al.*, *RoBERTa: A Robustly Optimized BERT Pretraining Approach*. 2019.
[3]J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," *CoRR*, vol. abs/1810.04805, 2018, [Online]. Available: http://arxiv.org/abs/1810.04805
[4]Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, *ALBERT: A Lite BERT for Self-supervised Learning of Language Representations*. 2020.
[5]J. Wei and K. Zou, *EDA: Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks*. 2019.
[6]P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, "Squad: 100,000+ questions for machine comprehension of text," *arXiv preprint arXiv:1606.05250*, 2016.
[7]J. Zhang, Y. Zhao, M. Saleh, and P. J. Liu, *PEGASUS: Pre-training with Extracted Gap-sentences for Abstractive Summarization*. 2020.
[8]C. Shorten, T. M. Khoshgoftaar, and B. Furht, "Text data augmentation for deep learning," *Journal of big Data*, vol. 8, no. 1, pp. 1–34, 2021.

For citations, source code, and more:

https://github.com/comacrae/noledge/