TASTY: TRANSFORMER-BASED AGGREGATE SUMMARY THROUGHOUT YELP

Micaela Siraj, Pedro H. R. Pinto, Praveen Balaji
Georgia Institute of Technology

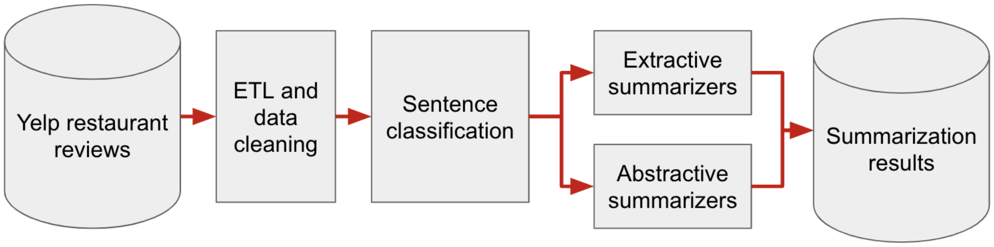
ABSTRACT

Formerly referred to as "Multi-Document Summarization for Review Analysis", the now *TASTY* paper seeks to prevent doomscrolling through restaurant reviews by generating review summaries with the most relevant information. *Doomscrolling* refers to the seemingly endless page scrolling while on a laptop or mobile device. Team 15 worked to develop an abstractive, classification summarizer that generates review summaries by both star and topic category. The goal is for the summaries to contain the most pertinent information from the overall group of reviews.

1 PROJECT OVERVIEW

Our team was able to develop an idea relatively fast by working through the 4 strategies that were provided to the class. We selected strategy 1, "Apply our learned models to solve problems in specific domains", to move forward. As a team, we were interested in text summarization techniques, specifically when it comes to reviews. The Yelp dataset stood out as it was easily accessible and could be focused in on food and restaurants. Below is an overview of the TASTY model.

Figure 1: Project overview



Restaurant reviews can vary greatly as the people who choose to write reviews can vary greatly. We first wanted to identify ways we could group the reviews to capture as much relevant information as possible without including outlier comments. Additionally we needed to identify summarization models that could capture the details from the grouped review inputs.

We found this project idea to be interesting due to it's immense practicality and ease of incorporation into everyday use cases, such as Yelp. With certain tweaks and modifications our TASTY model could be effectively applied to most any review based dataset.

2 LITERATURE REVIEW

From our research into text summarization, we identified the following categories: summarization through extractive models, summarization through abstractive models, summarization through both extractive and abstractive models and summarization through classification techniques. We also focused on looking at multi document summarization and summarizing short passages like tweets as we anticipated aggregating various restaurant reviews.

2.1 Summarization through extractive models

We identified 2 papers covering extractive methodologies. One of the papers also utilized the Yelp dataset (Goldenberg et al. 2017). The main focus of this paper was creating a tips-extraction framework that focused on pulling key words and phrases as helpful tips for users. The second extractive paper utilized BERT's text embeddings and then clustered the embeddings to identify sentences to select for a summary (Miller 2019). Our project utilizes Bert and key-word extraction, but goes even further to specify categories for more concise topic summarization.

2.2 Summarization through abstractive models

The abstractive based papers explored Yelp reviews, Amazon reviews, and multi-document summarization. The first paper we used developed a generative model for review summarization (Bražinskas et al. 2020). The challenge in this paper was tuning on novelty that would either benefit a summary or be irrelevant. The next paper also worked with reviews, but with a feed-forward neural network with attention-based encoder for abstractive summarization (Yang 2016). The last abstractive paper developed a multi-document abstractive summarizer, using ROUGE scores to improve summarization (Shapira and Levy 2020). Our team explored a couple of abstractive models in our development. We were interested in implementing ROUGE scores but determined it may be outside the scope of this project.

2.3 Summarization through both extractive and abstractive models

The survey paper we identified went through a comprehensive amalgam of both extractive and abstractive summarizers (El-Kassas et al. 2021). Our team also identified a paper on summarizing product reviews with both extractive and abstractive (Boorugu and Ramesh 2020). While extractive techniques can identify key sentences, these papers found abstractive summarizers to contain more quality context.

2.4 Summarization through classification, pre-training and domain adaptation

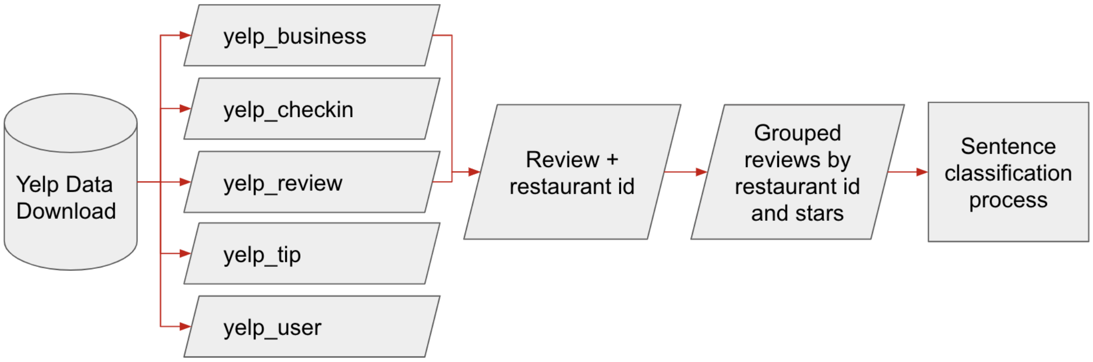
Where extractive and abstractive models can produce summaries, applying classification, pre-training and domain adaptation techniques can greatly enhance a model's output quality. (Gururangan et al. 2020) proved that large pretrained language models, such as BERT or GPT, can benefit from domain combining Domain Adaptation and Fine-Tuning for specific tasks. In this paper, the authors loaded a pretrained RoBERTa model, performed the original pretraining steps using four different domain-specific vocabularies, and were able to outperform the initial model on multiple classification tasks. A recent publication (Syed and Chung 2021) applied these techniques specifically to the restaurant review domain. The authors applied domain adaptation and fine-tuning to the BERT-base model for a Named Entity Recognition (NER) task and obtained similarly positive results. The last paper reviewed covered topic detection with summarization (Li et al. 2020). Our team worked to leverage several of these additional techniques to enhance our summarizer's performance.

3 Data Description

We were initially looking at both Amazon and Yelp reviews as possible datasets. We thought Amazon reviews may be to diverse and variant in its products and reviews. With Yelp, we could narrow in our subject to focus on specifically restaurant reviews.

Yelp data has 5 different data sets files containing different information in business identifiers. We joined the restaurant reviews file to the business ids file in order to aggregate reviews by restaurant. We then grouped these data together by business id and star rating before pushing them to the sentence classification process. The next page has an overview of the data ETL process.

Figure 2: ETL process overview



4 Proposed Models

4.1 Sentence Classification

In order to improve the quality of the review summaries, we decided to develop a model to classify each sentence of a review. The rationale was that by aggregating sentences that refer to similar ideas, it would improve the overall quality of the final summaries and provide a more useful user experience.

For this classification task, each sentence will be assigned to a single label. This task can be represented as choosing the class with the highest conditional probability given a sentence and the BERT model parameters (represented by the Greek letter θ).

$$Class = argmax(P(C|Sentence, \theta)) \tag{1}$$

Approached explored:

- BERT Domain Adaptation
- BERT Fine-tuning
- Weakly Supervised Learning

To come up with the different classes, we relied on a manual analysis of a small subset of Yelp reviews along with our own common sense knowledge as restaurant frequenters. We ended up choosing three classes to be used in the summarization process, as well as a fourth generic class to include all other types. A common example of sentences that fall in this last class, are the ones containing a non-specific statements about restaurants, such as "This restaurant was great!".

Classes:

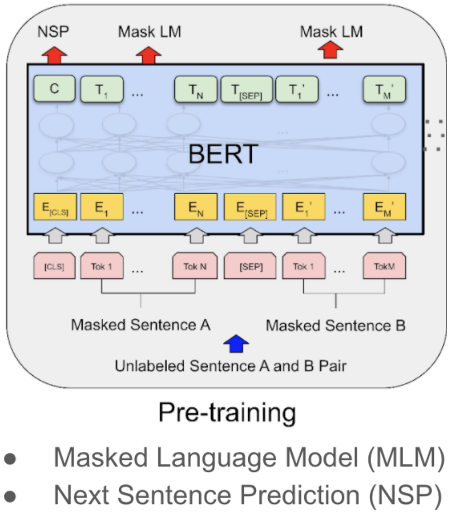
- Food Drinks
- Service
- Atmosphere
- Miscellaneous*

*The sentences labeled as “Miscellaneous” will not be used in the summarization process.

4.1.1 BERT Domain Adaptation

The original BERT model was trained using two simultaneous tasks: Next Sentence Prediction (NSP) and Masked Language Model (MLM). For the domain adaptation, we performed both of these tasks with 500,000 unique reviews from the Yelp dataset. Each review was converted into numerical tokens using the BERT tokenizer and were capped at 512 tokens. For the Masked Language Model, we masked 15% of each review with the special "mask" token. The full specifications of the pretraining setup are outlined in section 4.1.4.

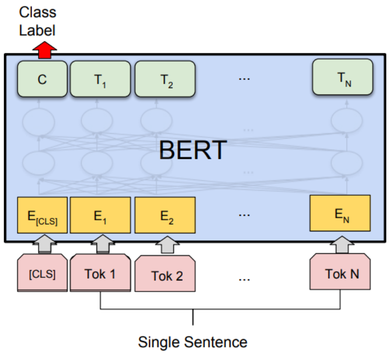
Figure 3: BERT pretraining process



4.1.2 BERT Fine-Tuning for Classificaiton

For the actual classification task, we fine-tuned a BERT based sequence classification model using individual sentences extracted from the Yelp dataset. We performed this process using the BERT-base model as well as the Domain Adapted BERT model generated on section 4.1.1. For training, the loss function used was the Categorical Cross Entropy and the optimizer was Stochastic Gradient Descent.

Figure 4: BERT fine-tuning setup for Sequence Classification



To obtain the predicted class for each sentence, we apply a softmax activation function to the output layer of the BERT based classifier and pick the class with the highest probability. The output of the *softmax* function represents the conditional probability $P(C|Sentence)$ presented on section 4.

$$\begin{aligned} Class &= argmax(P(C|Sentence, \theta)) \\ &= argmax\left(\frac{exp(c_i)}{\sum_j exp(c_j))}\right) \end{aligned} \tag{2}$$

In order to generate a training set for fine-tuning the BERT based classifier, we experimented with two different Weakly Supervised Learning techniques. This process is further explored in the next section.

4.1.3 Weak Supervision

Weak Supervision, also known as "Weakly Supervised Learning", revolves around using a less sophisticated, in other words “weak”, classifier to generate labeled data. This data is subsequently input into a more powerful classifier in the hopes that it can learn to generalize beyond the patterns detected by the weak one.

We explored two different Weak Supervision approaches to generate a labeled set to fine-tune the BERT based classifier: hand-crafted rules and using the Snorkel library.

Rule-based Approach:

The first step for this approach was to generate a lookup table with key words for each of our target categories. Using an external dataset combined with our common sense knowledge as restaurant frequenters, we compiled a table with the following distribution of keywords:

- Food/Drinks (F): 716 words
- Service (S): 105 words
- Atmosphere (A): 78 words

Subsequently, we devised the following sequential rules based on keyword counts:

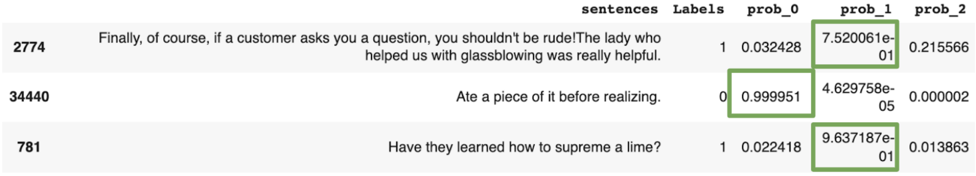
1. If 3 or more **A** words are present, label it as *Atmosphere*,
2. If 3 or more **S** words are present, label it as *Service*
3. If any category has at least 3 more key words than any other for a given sentence, assign that category to this sentence.
4. If a sentence only contains key words from a single category, and the keyword count represents at least 10% of the total number of words in that sentence, assign that label to the sentence.

Snorkel based Approach:

Snorkel is a Data-Centric artificial intelligence platform that provide features such as programmatic data labeling and weak supervision.

To leverage the *Snorkel* library in our project, we started by creating labeling functions combining keyword lookup as well as linguistic features generated by the *spaCy* library. Finally, we extracted probabilistic labels for each class and assigned each sample with the label containing the highest probability.

Figure 5: Snorkel output example

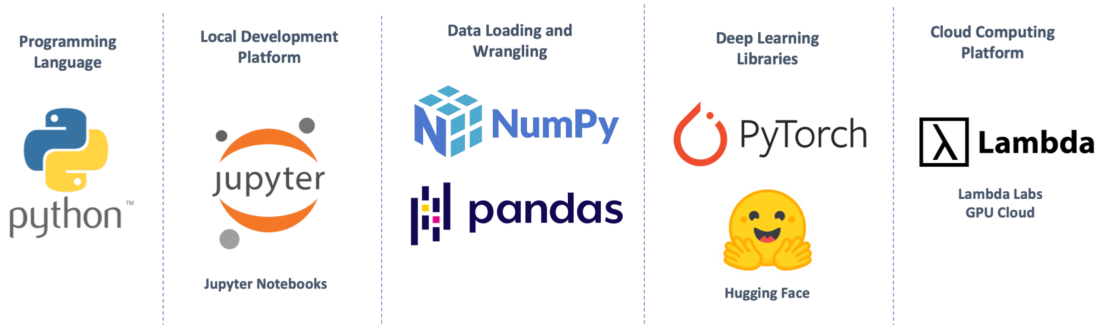


	sentences	Labels	prob_0	prob_1	prob_2
2774	Finally, of course, if a customer asks you a question, you shouldn't be rude!The lady who helped us with glassblowing was really helpful.	1	0.032428	7.520061e-01	0.215566
34440	Ate a piece of it before realizing.	0	0.999951	4.629758e-05	0.000002
781	Have they learned how to supreme a lime?	1	0.022418	9.637187e-01	0.013863

4.1.4 Implementation

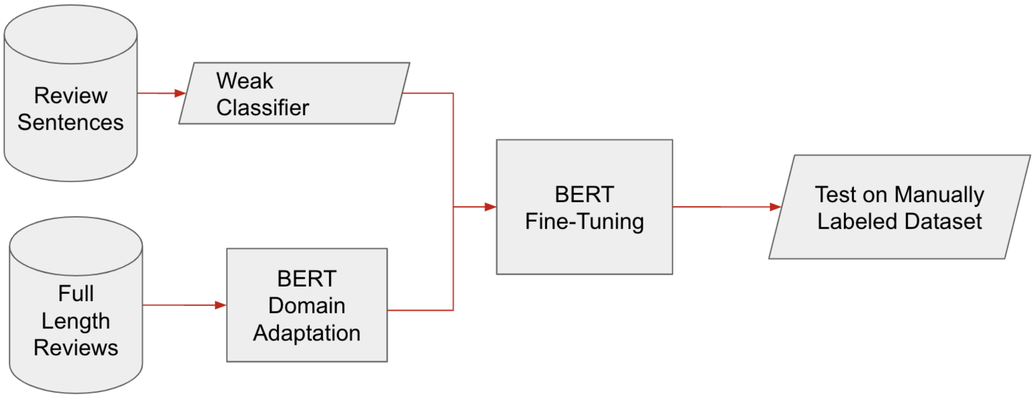
Python was the language used to implement the classification model and interactive *Jupyter* notebooks were used as the local development environment. The *Numpy* and *Pandas* libraries were used for data loading and wrangling while PyTorch was the chosen deep learning framework. We also utilized the *transformers* library developed by *Hugging Face* to import, train and test the BERT-base model. All the heavy computations were performed on NVIDIA GPUs hosted on the *Lambda Labs* public cloud.

Figure 6: Technology Stack used to implement the classification model



The full diagram for the proposed sentence classification approach is shown in Figure 6:

Figure 7: Proposed Approach (Sentence Classification)



4.1.5 Training Setup

BERT Domain Adaptation:

- Hardware:
  - 1X Nvidia A600 GPU (48GB), 14 vCPUs, 200 GB SSD
  - Hosted on the Lambda Labs Cloud
- Training Set:
  - 500,000 Yelp restaurant reviews (each review must contain at least two sentences to perform Next Sentence Prediction)
- Training Parameters:
  - 5 Epochs

    - Batch Size: 16
    - Learning Rate: 5E-4
- Training Time: 12:31:54 (hh:mm:ss)

Fine-tuning BERT for Sentence Classification:

- Hardware:
    - 1X Nvidia A600 GPU (48GB), 14 vCPUs, 200 GB SSD
    - Hosted on the Lambda Labs Cloud
- Training Set:
    - 100,000 sentences labeled with weak classifiers (using a lookup vocabulary): Hand-crafted rules:
        * Hand-crafted rules
        * Snorkel
- Test Set:
    - 1,095 manually labeled sentences
- Training Parameters:
    - 4 Epochs (as recommended on the original BERT paper)
    - Batch Size: 32
    - Learning Rate: 2E-5

4.2 Extractive Summarization

Our team tested out various extractive text summarization including Sumy, Pagerank, and Summarizer.sbert. Sumy is a python package that works to extract most relevant sentences. The pagerank algorithim can be used to rank most relevant sentences and return the selected top n sentences. The summarizer.sbert model comes from a python library that computes the cosine similarity across all possible sentence combinations. Where extractive models can retrun some of the most relevant sentences, some sentences may not always be relevant or make sense taken out of context of the entire passage. Our team found extractive models to somewhat defeats the purpose of a summarizer since it is not summarizing, but instead, cherry-picking sentences.

4.3 Abstractive Summarization

Our team also tested out various abstractive text summarization models. The primary hurdle in using abstractive models is the training data requires example summaries. This data does not exist, at least not in an accessible form and manually creating sufficient data is infeasible given the scale of the project. To this note, the models tested are pre-trained, typically with a news based data set, a potential factor in the results as the domains are vastly different. The first model tested is PRIMER, a multi-document text summarization model. This model was tested as it would allow summarization of different reviews without first merging them or possibly even classifying, though the results were underwhelming in initial tests and we did not move forward with its use. Additionally, two more traditional summarization models, GPT-2 and T5 were tested (post classification, with sentences appended to form one large document), with decent results. Subjective evaluation showed that T5 produced more coherent summaries, and as such was the model used in the final project.

5 Results

The final pipeline was tested on a subset of the Yelp restaurant review dataset, with 100,000 sentences. Additionally, examples of summaries produced by the final model are provided from the 1063 Google Maps reviews for the Rocky Mountain Pizza Company.

5.1 SENTENCE CLASSIFICATION RESULTS

For the sentence classification task, we tested the two BERT based models (BERT-base and Domain Adapted BERT) combined with the two Weak Supervision methods (Rule-based and Snorkel). These models were tested with 1,095 manually labeled sentences extracted from Yelp restaurant reviews. There were the results:

- BERT-BASE + Rule Based Weak Supervision:
  - Validation Accuracy: 99%
  - Test Accuracy: 73.51%
- Domain Adapted BERT-BASE + Rule Based Weak Supervision:
  - Validation Accuracy: 99%
  - Test Accuracy: 73.34%
- BERT-BASE + Snorkel:
  - Validation Accuracy: 98%
  - Test Accuracy: 51.05%
- Domain Adapted BERT-BASE + Snorkel:
  - Validation Accuracy: 50%
  - Test Accuracy: 51.05%

It is important to note that the accuracy measure includes the unwanted “Miscellaneous” class, which tends to be harder to classify. Additionally, low-quality reviews tend to receive less confident predictions.

Based on our tests, neither the domain adaptation or the Snorkel based Weak supervision yielded in accuracy improvements. However, both rule-based BERT models outperformed simply applying the Rule-Based approach to the test set. The hand-crafted rules to the set resulted in an accuracy measure of **52.51**%. This is positive evidence that the model is being able to generalize the linguistic patterns of each type of sentence.

Given these results, the BERT-base + Rule Based Weak Supervision was used to classify the sentences for the downstream summarization task. To ensure high-confidence classifications, we only selected sentences whose assigned class had a probability of at least 85%

5.2 EXTRACTIVE SUMMARIZATION RESULTS

Listed below are the 3 different extractive methods we used an example of the outputs from each. We found that summy overall provided better extractions. Pagerank and sbert occasionally returned the same sentences depending on the n selected to be returned. The examples below show pagerank and sbert returning the same extracted sentences.

- Sumy:“Great music, friendly staff, nice regulars The food: best mozzarella sticks I've ever had. I ordered the cajun, smokehouse, mac and cheese bites and onion rings and everything tasted great! RM has the best nachos I've had no question about it their food is spot on and fair priced, and their service is always great thank you for the card guys.”
- Pagerank: “The wings were delicious and I was surprised how good the pizza was. Pizza was very good and muffuletta was fantastic! We will be back! This is my home away from home to catch my PHILADELPHIA EAGLES play. Great service, however the food took a little to long to come out.”
- Summarizer.sbert:"The wings were delicious and I was surprised how good the pizza was. Pizza was very good and muffuletta was fantastic! We will be back! This is my home away from

> home to catch my PHILADELPHIA EAGLES play. Great service, however the food took a little to long to come out."

5.3 Abstractive Summarization Results

Example results for the abstractive summarization following the classification process are shown in the table below.

Sentiment	Food	Service	Atmosphere
Positive	the wings were delicious and i was surprised how good the pizza was. my partner said his calzone was on point. dr.robot sour beer was the best we’ve had since moving down here, the closest to home that we’ve been able to get. it was very good and the food was really good. great food and drinks. the beer is cheap too and they have a good selection of draft beers.	austin is the best bartender ever!bartender ben rocks,carlos is hilarious! the entire staff is so kind and respectful. parking can be pretty bad when it’s busy, but a nice place to walk or ride share to. nice atmosphere and good food, i cannot wait to go back very friendly staff. the servers are amazingly accommodating and friendly. if you come, ask for lucy!	i ordered take out. a feel good place to go with friends, eat pizza and wings while talking about life. easy access to georgia tech campus, friendly atmosphere, good food. the environment is awesome. have inside as well as outside sitting. the food and atmosphere are great as long as you don’t get your credit card stolen,it has a great ambiance, just below average food outside of pizza.
Negative	pizza is mediocre,i chalked it up to it being a thin crust pizza. pizza dough is frozen and chewy. tasted like microwave. food is deplorable, but it’s a college bar so that’s not why i give it 2starz. half the chicken sandwich didn’t have much chicken. there are better sports bars in atlanta.	the service was poor and really slow. it’s vicinity to gt makes this pizza joint a college staple for students but its quality and service need serious improvement. the manager was aggressive and completely unprofessional. never got asked if we needed anything else or we’re okay. he said that people were listening to the juke box, which is fair, but after the (crappy songs were done playing... he didn’t bother to bring drinks to our table.	philadelphia eagles bar, pizza is good, great atmosphere during football season. the decor was drab and generic without any flair. hostess misinformed us and sat us in a reserved seating area for someone else. i’m hungry and again don’t feel like going far, so let’s give them one last chance. it’s great as long as you don’t get your credit card stolen, but it is very likely that you will.

As can be seen, the results with positive reviews tend to be more coherent and on topic. This is likely a result of two factors. One, there are fewer negative reviews for restaurants, especially one like rocky mountain pizza (rated 4.2*), giving fewer reviews to work with. Two, negative reviews often don’t comment on the same thing, but rather a unique event that made an experience bad, furthermore, they often don’t use proper grammar, use special characters like emojis, etc, all of which make summarizing more difficult. In extreme situations with low data and bad filtering, abstractive summarization can create false summaries, as exemplified by the following output, where it makes it seem as though a manager is stating what reviewers posted:

```
"Place wasn't busy," says a mgr. "the kitchen is closing,"
he tells the waitress the pizza was unacceptable - "it looked fine to me"
"i will make sure I tell everyone not to patronize this business as well!"
```

This outcome is not desirable, and as such discerning between when to use abstractive and extractive summarization is important future work.

6 Analysis and Key Takeaways

The abastractive models were able to take away the most relevant content from tested examples to create newly generated (abstracted) text as a true summary. We were also able to show that classifying sentences can aid in summarization by aggregating related content by class. The models generated more coherent and relevant summaries by focusing on one class at a time

However, a major challenge when performing summarization without a labeled dataset, is coming up with an objective evaluation of the results. more time and research could help improve the analysis of results and develop better metrics for unsupervised summarization.

We also realized that a significant amount of processing power and time is needed in order to experience the benefits of domain adaptation for large language models like BERT. Based on our results pretraining the BERT model for 5 epochs with 500,000 was not enough to adapt our classifier for the restaurant review domain. It is safe to assume that, if we had the resources to perform domain-specific pretraining for longer (such as for 100 epochs following (Gururangan et al. 2020)), we should expect better results.

Training classifiers with limited data proved to be quite a challenging task. However, incorporating techniques like Weak Supervision and Transfer Learning proved effective for our project. Unfortunately using the Snorkel library did not improve our accuracy measures. This probably means that Snorkel is either not the right tool for our use case or more work needs to be performed in preparing and fine-tuning it for our task.

7 Conclusion and Future Work

Overall, we were to achieve our goal of coming up with a solution to prevent doomscrolling through online restaurant reviews. By combining sentence classification with summarization, our multi-step model was able to generate high-quality relevant summaries that could benefit real-world users of websites like Yelp, Foursquare or Tripadvisor if ever deployed to production.

During this process, we were able to put in practice several of the topics learned in class, such as transformer-based language models, pretraining procedures, weak supervision, text generation, among others. We combined these different models and techniques to explore to a novel task in the domain of restaurant reviews.

For future work in the classification task, we could experiment with other techniques like Semi-Supervised learning and Active Learning to improve accuracy as well as crowd-source the manual labeling exercise to increase our test set. We could also invest in pretraining the BERT model for more epochs to truly experience the benefits of domain adaptation. For the summarization task we can continue to experiment with other abstractive methods while trying to develop better accuracy measures for unsupervised summarization.

References

[Boorugu and Ramesh 2020] Boorugu, R. and Ramesh, G. (2020). A survey on nlp based text summarization for summarizing product reviews. *2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA)*, pages 352–356.

[Bražinskas et al. 2020] Bražinskas, A., Lapata, M., and Titov, I. (2020). Unsupervised opinion summarization as copycat-review generation. pages 5151–5169.

[El-Kassas et al. 2021] El-Kassas, W. S., Salama, C. R., Rafea, A. A., and Mohamed, H. K. (2021). Automatic text summarization: A comprehensive survey. *Expert Systems with Applications*, 165:113679.

[Goldenberg et al. 2017] Goldenberg, D., Kampel, G., and Cohen, Y. (2017). Needle in a haystack: Tips extraction from yelp reviews.

[Gururangan et al. 2020] Gururangan, S., Marasovic, A., Swayamdipta, S., Lo, K., Beltagy, I., Downey, D., and Smith, N. A. (2020). Don't stop pretraining: Adapt language models to domains and tasks. *CoRR*, abs/2004.10964.

[Li et al. 2020] Li, P., Huang, L., and Ren, G. (2020). Topic detection and summarization of user reviews. *CoRR*, abs/2006.00148.

[Miller 2019] Miller, D. (2019). Leveraging BERT for extractive text summarization on lectures. *CoRR*, abs/1906.04165.

[Shapira and Levy 2020] Shapira, O. and Levy, R. (2020). Massive multi-document summarization of product reviews with weak supervision. *CoRR*, abs/2007.11348.

[Syed and Chung 2021] Syed, M. H. and Chung, S.-T. (2021). Menuner: Domain-adapted bert based ner approach for a domain with limited dataset and its application to food menu domain. *Applied Sciences*, 11(13).

[Yang 2016] Yang, L. (2016). Abstractive summarization for amazon reviews.