Span Extraction using Transformers

Arvind Reddy Bobbili
abobbili@cougarnet.uh.edu

Abstract

This paper introduces a novel approach to aspect-based sentiment analysis (ABSA) through the application of advanced transformer models—T5, BART, and Pegasus—for sentiment span extraction. ABSA is critical for deriving nuanced consumer insights as it identifies sentiments related to specific aspects within texts. Our innovative methodology involves training these transformer models on a diverse, annotated dataset, significantly enhancing their ability to precisely recognize sentiment expressions. Subsequently, the models are fine-tuned on a targeted dataset designed specifically for extracting sentiment spans, thus achieving high precision in identifying sentiment-related text spans. The performance of each model is evaluated using metrics such as the Jaccard Index and BLEU Score, which confirm their effectiveness in generating accurate and contextually relevant sentiment spans. The results demonstrate that our novel approach not only improves extraction accuracy but also extends the capabilities of current natural language processing models in specialized sentiment analysis tasks, offering substantial potential benefits across various consumer-facing industries.

1 Introduction

Aspect-based sentiment analysis (ABSA) is a crucial component of natural language processing (NLP), as it enables us to delve into the nuances and emotions conveyed in text. This process involves identifying and examining the sentiment expressed toward specific aspects or features of a product, service, or topic. Such detailed analysis provides us with valuable insights that may be overlooked by traditional sentiment analysis methods. For instance, a restaurant with predominantly positive reviews may still have average ratings overall. This discrepancy could arise from positive sentiment toward the food quality but negative sentiment toward the service. This is where aspect-based sentiment analysis proves invaluable. Our primary objective is to extract these aspects from user reviews to effectively conduct this analysis.

Transformers, a class of models that have revolutionized the field of NLP, offer significant advantages in handling complex tasks like ABSA due to their ability to capture contextual relationships in text. This paper explores the application of advanced transformer models such as T5, BART, Pegasus for the task of sentiment span extraction. This process involves precisely extracting the text span that pertains to a particular aspect.

2 Literature Review

This task shares similarities with various information extraction problems. Previous works have employed techniques such as supervised keyphrase extraction [1], and re-ranking approaches [2]. Additionally, segment features via semi-CRFs [3, 4] and syntactic feature-based approaches [5, 6, 7, 8] have been successful in opinion mining. Machine Learning Models like SVMs have been utilized in sentiment span extraction to classify segments of texts as expressing positive, negative, or neutral sentiments based on feature vectors derived from the text [9]. Conditional Random Fields (CRFs) have been a popular choice for sequence labeling tasks, including sentiment span extraction, due to their ability to model the conditional probability of a label sequence given a particular sequence of observations [10]. My work is greatly influenced by Mukherjee's [11, 12] work. Utilizing the same dataset as theirs, I introduce a novel approach using transformers, a notably new model.

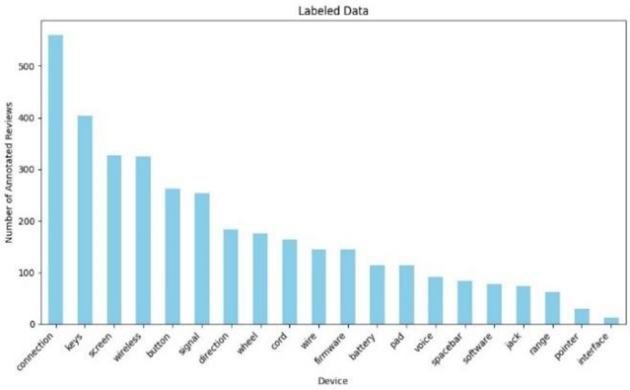

Figure 1 – Labeled Data

Additionally, I am exploring machine learning models in transformers that, to the best of my knowledge, have not been previously utilized. This methodology represents a fresh perspective, potentially offering performance comparable to or surpassing existing methods.

3 Data

3.1 Raw Data - Files

In this study, we utilized a diverse dataset curated specifically for aspect-based sentiment analysis. The dataset comprises a wide range of 1,2 3-star reviews from amazon.com. The dataset encompasses multiple domains. It is labeled data across 6 domains, namely earphones, GPS, keyboards, mouse, MP3 player, and router. Each of these folders contains multiple.txt files named according to aspects such as cord, jack, screen, pad, etc. which contain aspect-specific reviews.

All the reviews in the .txt files have review sentences followed by verbs and adjectives at the end of the sentence in [VBZ, VBD tags] and adjectives or adverbs [ADJ, ADV tags]. Out of all these instances, few of them are annotated, meaning the head aspect/Issue is marked in the sentence. The <i> tag is used to indicate the beginning and end of the head aspect of the review sentence, and this annotation process was done by the domain experts.

There are approximately 12,096 reviews in total, with 3,583 instances annotated and 8,513 instances unannotated. This means that 29.7 percent of the data is annotated, which is a considerable amount suitable for training a transformer model. Transformer models are typically pre-trained on vast amounts of data, and while 29.7 percent may seem modest, it should suffice to effectively train models like T5, BART, and Pegasus.

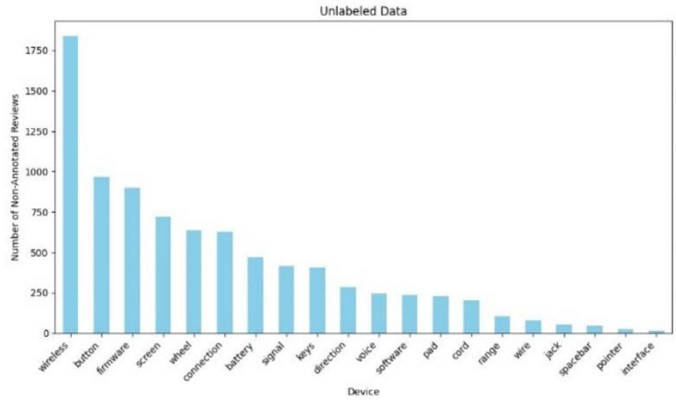

Figure 2 – Unlabeled Data

Upon conducting exploratory data analysis (EDA), I noticed that certain aspects, such as "connection" and "wireless," have a significantly higher number of instances (As observed in Figure 1 and Figure 2). Additionally, aspects like "buttons" are prevalent in both mouse and MP3 player reviews, while "jack" is commonly mentioned in reviews for earphones and MP3 players.

3.2 Preprocessing

For convenience, I have chosen to partition the dataset into labeled and unlabeled subsets. The labeled dataset will comprise 3 columns – review, aspect, and span (this is extracted from the review, the portion of the sentence that is enclosed within `<i>` tags. Also, I am not using ADJ, ADV and verb tags. The unlabeled dataset contains 2 columns with one column containing sentences and the other column containing aspects. The labeled data is now split into training and validation sets with 80:20 ratio. Training dataset size – 2865 records, Validation dataset size – 717 records. We then append a fixed prefix, like, "find span in the sentence: ", to each review sentence to create a prompt-like structure that guides the model to focus on span extraction. These prefixed sentences are then tokenized using a tokenizer object's tokenizer() method, with a maximum sequence length of 1024 tokens and truncation enabled to handle inputs longer than this length. Similarly, the target spans are tokenized separately, also with truncation and a shorter maximum length of 128 tokens.

4 Methods

I am utilizing advanced natural language processing models, namely T5 (Text-To-Text Transfer Transformer), BART (Bidirectional and Auto-Regressive Transformers) and Pegasus to perform ABSA. These state-of-the-art models have demonstrated exceptional performance across a wide range of natural language understanding tasks.

4.1 T5 Transformer

The Text-To-Text Transfer Transformer (T5) model [13], introduced by Google AI researchers, represents a significant breakthrough in the field of natural language processing (NLP). Unlike traditional NLP models that are tailored for specific tasks, T5 adopts a unified framework capable of handling a diverse range of NLP tasks through a text-to-text approach.

The T5 model, renowned for its versatility and performance across various natural language processing (NLP) tasks. T5's architecture, based on the Transformer framework, is characterized by its encoder-decoder structure, where the encoder processes the input text, and the decoder generates the output text. This architecture enables T5 to handle both autoregressive and non-autoregressive tasks efficiently, making it suitable for a wide range of NLP Applications.

The T5 model is famous for language translations, but I leveraged this and framed the task as a text-to-text transformation where the input text is a review sentence, and the output text is the sentiment expression span corresponding to that aspect.

For example, input is “The spacebar is so bad, the product is completely unusable for me” and the output would be "Spacebar is so bad”.

We already have a dataset of labeled aspect-specific sentiment expressions where we have sentences in one column and the span of those sentences in one more column. This dataset will serve as the training data for the T5 model, like the input-output pairs. I have fine-tuned the pre-trained T5 model on the labeled dataset using a sequence-to-sequence approach. The model learns to map input review sentences to output sentiment expression spans. I have trained the model for 6 epochs with a learning rate of 0.00001 with a batch size of 16, as I noticed the model is leading to overfitting later. parameters to minimize the discrepancy between predicted and ground truth sentiment expression spans.

4.2 BART

BART (Bidirectional and Auto-Regressive Transformers) has emerged as a powerful model for various natural language processing tasks [14]. ABSA involves identifying aspects or features mentioned in the text. BART's unique architecture and capabilities make it particularly well-suited for ABSA tasks. At its core, BART is based on the Transformer architecture. BART uses a standard seq2seq/machine translation architecture with a bidirectional encoder (like BERT) and a left-to-right decoder (like GPT). What sets BART apart is its bidirectional nature, allowing it to efficiently encode and decode text in both directions.

In the context of ABSA, BART's bidirectional capabilities are particularly advantageous. By considering the entire input text bi-directionally, BART can effectively capture the relationships between aspects and their corresponding sentiments. This made me feel like BART can generate more accurate and contextually informed sentiment predictions for each aspect.

Moreover, BART's auto-regressive nature allows it to generate output tokens autoregressively, meaning each token is generated based on previously generated tokens. This sequential generation process enables BART to produce fluent and coherent text, making it well-suited for our tasks where the output needs to be linguistically coherent.

I fine-tuned the pre-trained BART model on the labeled dataset which was prefixed with commands using a sequence-to-sequence approach for 6 epochs, with a learning rate of 0.0001 and a batch size of 32 and 16. The batch size 16 has given better results.

4.3 Pegasus

Pegasus (Pre-training with Extracted Gap-sentences for Abstractive SUmmarization), a state-of-the-art model in natural language processing, has shown promising results in various text generation tasks [15]. Unlike traditional models that focus on generating text sequentially, Pegasus employs a novel approach known as pretraining with reconstruction objectives, which enables it to capture the underlying structure and semantics of the input text. Its pre-training involves using an approach called "self-supervised objective GAP sentences generation." In this technique, certain sentences are masked (removed) from the input document during training, and the model is trained to generate these missing sentences from the rest of the text.

While Pegasus is initially designed for summarization task, I have fine-tuned on the labeled dataset using a sequence-to-sequence approach During fine-tuning, Pegasus learns to give output that is present inside the input sentence such that it is around the aspect.

This model has taken the utmost time and computational resources as it is very complex and robust. I have trained the model for 5 epochs and I have tweaked the learning rate to 0.0002. The batch size for training and validation dataset being 16.

5 Evaluation

We are going to get the labels and predictions from the model and decode them to perform evaluation. evaluating the similarity between two sets. In the context of aspect

5.1 Jaccard Index

The Jaccard Index is a widely used metric for evaluating the similarity between two sets. In the context of aspect-specific sentiment expression spans, it provides a quantitative measure of the overlap between predicted spans generated by the model and the ground truth spans from the labeled dataset. The formula for calculating the Jaccard Index is given by:

$$\boldsymbol{J(A,B)} = \frac{\boldsymbol{A \cap B}}{\boldsymbol{A \cup B}}$$

Where A represents the set of tokens in the predicted span, B represents the set of tokens in the ground truth span, $|A \cap B|$ denotes the size of the intersection between A and B, and $|A \cup B|$ denotes the size of the union of A and B.

A higher Jaccard Index value indicates a greater overlap between the predicted and ground truth spans, indicating better performance of the model in identifying aspect-specific sentiment expressions.

5.2 Bleu Score

The BLEU (Bilingual Evaluation Understudy) Score [16] is commonly used in machine translation tasks. But I am trying to evaluate the

$$\text{BLEU} = \min\left(1, \frac{\text{output-length}}{\text{reference-length}}\right)\left(\prod_{i=1}^{4} \text{precision}_i\right)^{\frac{1}{4}}$$

quality of aspect-specific sentiment expression spans generated by the model compared to the ground truth spans. The formula for calculating the BLEU Score involves precision, defined as follows:

The ratio output-length / reference-length compares the length of the predicted span text to the reference text.

5.3 Average Generation Length

The Average Generation Length is a metric used to evaluate the length of the aspect-specific sentiment expression spans generated by the model. It measures the average number of tokens in the predicted spans across all predictions. The formula for calculating the Average Generation Length is straightforward:

$$Avg\ Length = \left\{ \sum_{\{i=1\}}^{N} \{Length\}_i \right\}$$

Where N is the total number of predictions, Length_i represents the length of the predicted span for the *ith* prediction. A longer average generation length may suggest that the model is generating more detailed or informative spans, but it could also indicate the generation of irrelevant or verbose spans.

5.4 Manual Evaluation

As the test data of around 8,000 records is unlabeled. I could not perform any evaluation, as I couldn't find any metric as after significant research. Hence, I chose to manually verify if the span was generated as expected. Though I cannot accurately quantify my findings, upon looking at the predictions, I felt all the models did give a fair performance and extracted the span related to the aspect, for most records.

6 Results

Model/Metric	Training loss	Validation loss	Bleu Score	Jaccard Index	Avg Gen length
T5	0.2278	0.2696	83.3955	0.8567	5.7183
BART	0.1581	0.2496	87.7594	0.8922	8.2120
Pegasus	0.1410	0.2306	83.8838	0.8628	6.2036

Based on the results obtained from running the models for the problem statement mentioned above, several key observations can be made. A slightly regarding the training and validation losses, we can see that BART achieved the lowest training loss of 0.1581, followed closely by Pegasus with a loss of 0.1410, and T5 with a slightly higher loss of 0.2278. However, in terms of validation loss, BART still maintains its lead with a loss of 0.2497, indicating its robustness in generalizing to unseen data. Pegasus follows closely with a validation loss of 0.2307, while T5 exhibits a slightly higher loss of 0.2697.

Moving on to evaluation metrics, both BLEU Score and Jaccard Index provide insights into the quality of generated sentiment expression spans. BART achieves the highest BLEU Score of 87.7595, indicating a high level of similarity between its predictions and the ground truth spans. Pegasus follows closely with a BLEU Score of 83.8838, while T5 trails behind with a score of 83.3955. Similarly, BART also outperforms the other models in terms of Jaccard Index, achieving a score of 0.8922, reflecting a higher degree of overlap between its predictions and the ground truth spans. Pegasus follows with a Jaccard Index of 0.8628, while T5 lags slightly behind at 0.8567.

Finally, considering the average generation length, BART produces longer sentiment expression spans with an average length of 8.212 tokens. Pegasus generates spans with an average length of 6.2036 tokens, while T5 produces shorter spans on average with a length of 5.7183 tokens. These results suggest that while BART exhibits superior performance in terms of loss, evaluation metrics, and average generation length, Pegasus also demonstrates competitive performance across these metrics. However, further analysis and experimentation may be warranted to gain deeper insights into the strengths and weaknesses of each model for the specific problem statement.

Overall, I was successfully able to achieve significant results by tweaking the parameters for different models. I have also shown and learnt how we can fine-tune models created for some purposes like translation, summarization etc., can be used for span extraction.

7 Limitations

There have been challenges during the development and deployment of models. One such challenge is limited data availability. While starting with 29% of labeled data is a good place to begin, I believe that having access to more data for model training could lead to even better performance. Increasing the dataset size can enhance the model's ability to capture diverse patterns and nuances present in the data, potentially resulting in improved accuracy and generalization. Therefore, expanding the available data resources could be a valuable strategy for overcoming the limitations posed by data scarcity in machine learning tasks. Additionally, achieving the optimal balance between model complexity and generalization typically demands substantial experimentation and computational resources. While I believe I've made diligent efforts and attained good results I still feel there remains potential for enhancement. As the test data is unlabeled, there is no quantifiable evidence to show accuracy. The model cannot be generalized to any reviews, as the dataset it is trained on, particularly contains only gadget related reviews.

8 References

[1] Berend G (2011) Opinion Expression Mining by Exploiting Keyphrase Extraction. In: Int. Jt. Conf. Nat. Lang. Process. pp 1162–1170

[2] Johansson R, Moschitti A (2010) Reranking models in fine-grained opinion analysis. Proc 23rd Int Conf Comput Linguist 519–527.

[3] Yang B, Cardie C (2012) Extracting opinion expressions with semi-markov conditional

random fields. In: Proc. 2012 Jt. Conf. Empir. Methods Nat. Lang. Process. Comput. Nat. Lang. Learn. pp 1335–1345

[4] Klinger R, Cimiano P (2013) Bidirectional Interdependencies of Subjective Expressions and Targets and their Value for a Joint Model. Assoc. Comput. Linguist. (Short Pap.

[5] Kim S-M, Hovy E (2006) Extracting opinions, opinion holders, and topics expressed in online news media text. Proc Work Sentim Subj Text, Assoc Comput Linguist 1–8.

[6] Wu Y, Zhang Q, Huang X, Wu L (2009) Phrase dependency parsing for opinion mining. Proc 2009 Conf Empir Methods Nat Lang Process 1533–1541.

[7] Jakob N, Gurevych I (2010) Extracting opinion targets in a single- and cross-domain setting with conditional random fields. Proc 2010 Conf. Empir Methods Nat Lang Process 1035–1045.

[8] Kobayashi N, Inui K, Matsumoto Y (2007) Extracting Aspect-Evaluation and Aspect-Of Relations in Opinion Mining. Proc. 2007 Jt. Conf. Empir. Methods Nat. Lang. Process. Comput. Nat. Lang. Learn. (EMNLP-CoNLL)

[9] Pang, B., & Lee, L. (2004). A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics.

[10] Jiang, L., Yu, M., Zhou, M., Liu, X., & Zhao, T. (2011). Target-dependent Twitter sentiment classification with rich automatic features. Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies.

[11] Li H, Mukherjee A, Liu B, Si J (2015) Extracting Verb Expressions Implying Negative Opinions. Proc. twenty-ninth AAAI Conf. Artif. Intelligence

[12] Arjun Mukherjee - Extracting Aspect Specific Sentiment Expressions implying Negative opinions

[13] Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., & Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. Journal of Machine Learning Research, 21, 1-67.

[14] Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., & Zettlemoyer, L. (2019, October 29). BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension.

[15] Zhang, J., Zhao, Y., Saleh, M., & Liu, P. J. (2020). PEGASUS: Pre-training with extracted gap-sentences for abstractive summarization. Proceedings of the 37th International Conference on Machine Learning, 2020.

[16] Papineni, K., Roukos, S., Ward, T., & Zhu, W.-J. (2002). BLEU: a Method for Automatic Evaluation of Machine Translation. In Proceedings of the 40th Annual Meeting on Association for Computational Linguistics (pp. 311-318). Philadelphia, PA: Association for Computational Linguistics.