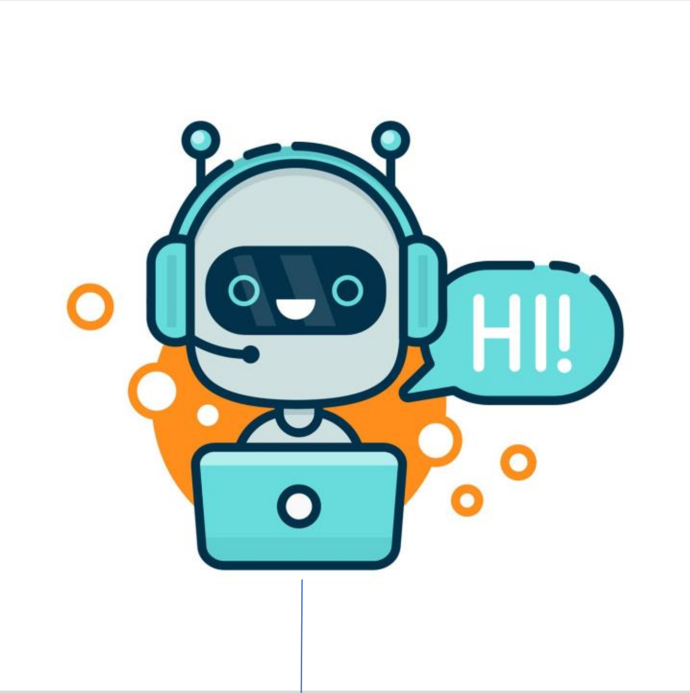
Using GPT-2,GPT-NEO, T5 to classify Explanations of Movie Recommendations as Good or Bad via a novel quality metric

Presenter: Joseph May

Date: 4/1

Background: Conversational Recommender Systems

Chatbots



Recommender system

Large Language Models

ChatGPT

CRS

Search Vs. CRS

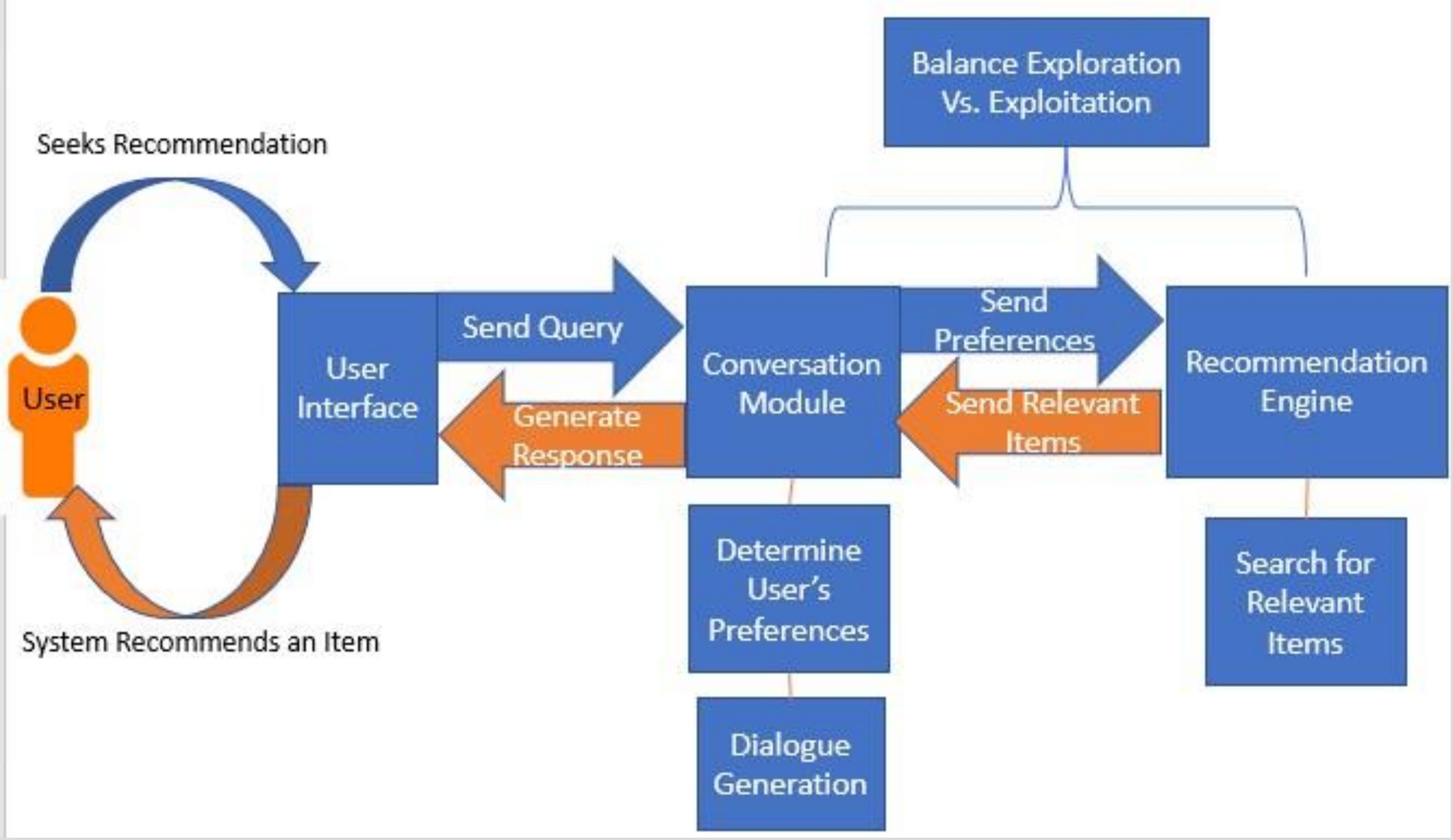
Search

- Keywords
- Single Interaction
- Specific

CRS

- Natural Language
- Multi-round
- Exploratory
- Feedback
- Context window
- Sidestep cold start problem
- Avoid user drift over time

CRS: How Do They work?



Dialog Strategy: How does the Conversation Feel?

- Active User Passive System (AUPS):
  - System only responds to direct user prompts
  - Search Engine / Voice Assistant
- System Active User Passive (SAUP):
  - User responds to system
  - User does not volunteer info other than initial prompt
  - System interrogates user
- System Active User Engage (SAUE):
  - System engages user
  - User may respond and provide additional feedback
  - Chit-chat capable
  - Formal conversation between two humans
- System Active User Active (SAUA):
  - System engages user
  - User may alter and direct conversation
  - Two humans conversing informally

Lower Complexity

Higher Complexity

Evaluation Patterns in Other Papers

1

Create CRS

- Recommendation Engine
- Conversation Engine
- Test and report metrics (NDCG, MRR, BLEU etc)

2

Online user survey: Author model versus other model(s)

3

Survey Comparison Issues

- Relativity
- Comparing restaurants

How to Evaluate System Performance?

- Turn Level: Evaluate each sentence
- Dialogue Level: Evaluate the whole

Conversational Quality

- conversation
- BLEU
- ROGUE
- METEOR
- MAUDE
- Readability
- Novel Sentence Evaluation
- Perplexity

Recommendation Quality

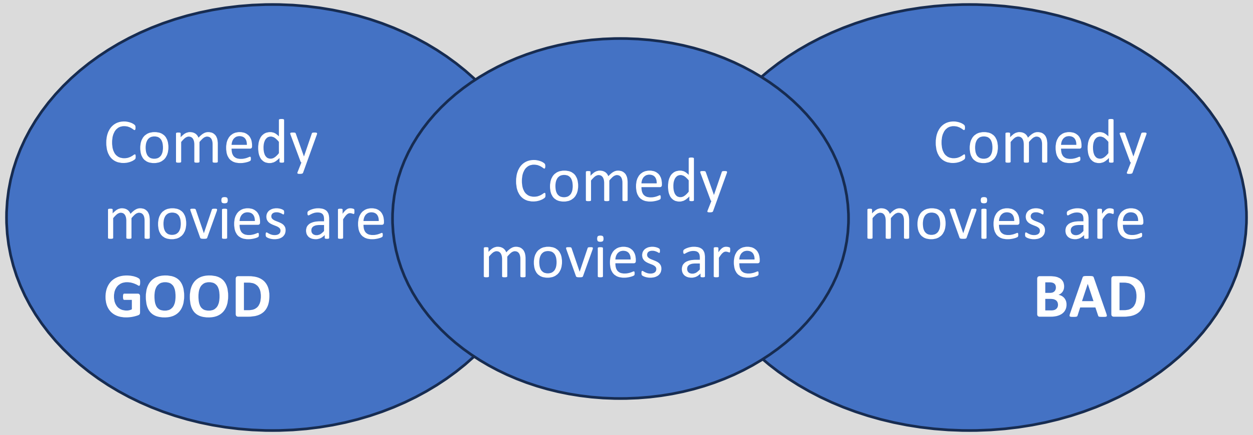
- Precision
- Recall
- Normalized Discounted Cumulative Gain
- Mean Reciprocal Rank
- Coverage
- Personalization

Quality Gaps

Assessment Metrics

- BLEU
- ROGUE
- METEOR
- Perplexity
- Deep Learning Regimes
  - BERT
  - ChatGPT
  - Transformers
  - Contrastive Learning
  - Word embeddings

Word Overlap



Translation Accuracy

1. The cat ran fast
2. The animal moved hastily
3. The beast moved

How to Evaluate System Performance?

Conversational Quality

- Assume the recommendation engine exists.
- Focus solely on evaluating conversation engine.
- Offline evaluations

Recommendation Quality

- [illegible]ision
- N[illegible]
  Cum[illegible]
- M[illegible]
  [illegible]e
- [illegible]sonalization

What Makes A Good recommendation?

Factors of Explainability	Definition
Relevance	If the recommendation is relevant to the query
Length	How long the explanation is
Readability	How easy the recommendation is to read
Word Importance	The importance of words in the recommendation
Repetition	How many duplicate segments are in a sentence
Subjectivity	If the recommendation includes personal opinions and emotion
Polarity	Confidence level that the recommendation is positive or negative
Grammatical Correctness	Misspelled words and incorrect usage of language
Feature Appearance	If an explanation captures item features

Quality can be subjective!

User: I'm looking for a fun movie with Samuel L. Jackson in it, or a movie with cool gadgets.

Try Captain America Winter Soldier

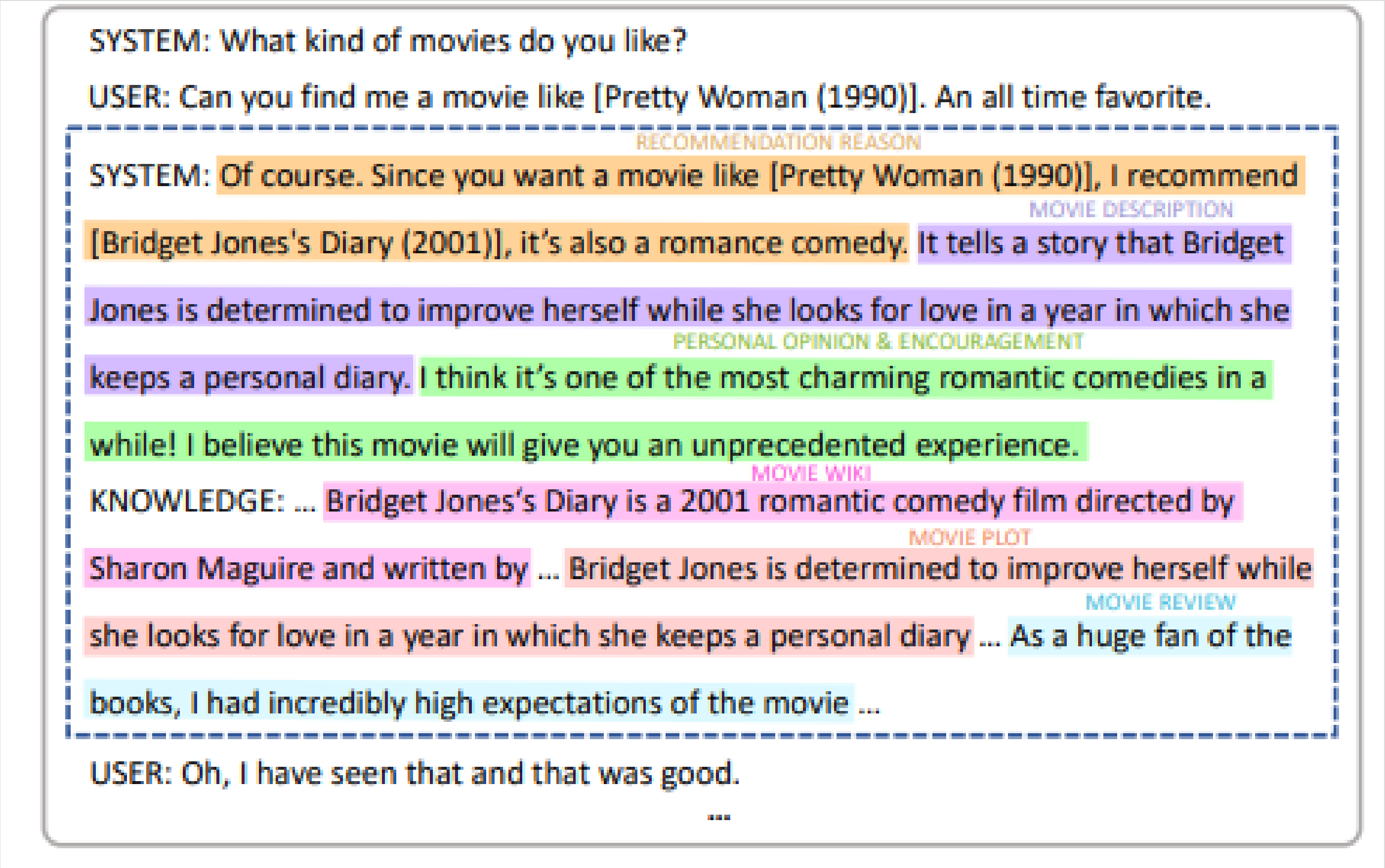
Try Spiral (Saw 9)

Dataset: E-Redial & INSPIRED

Dataset Information:

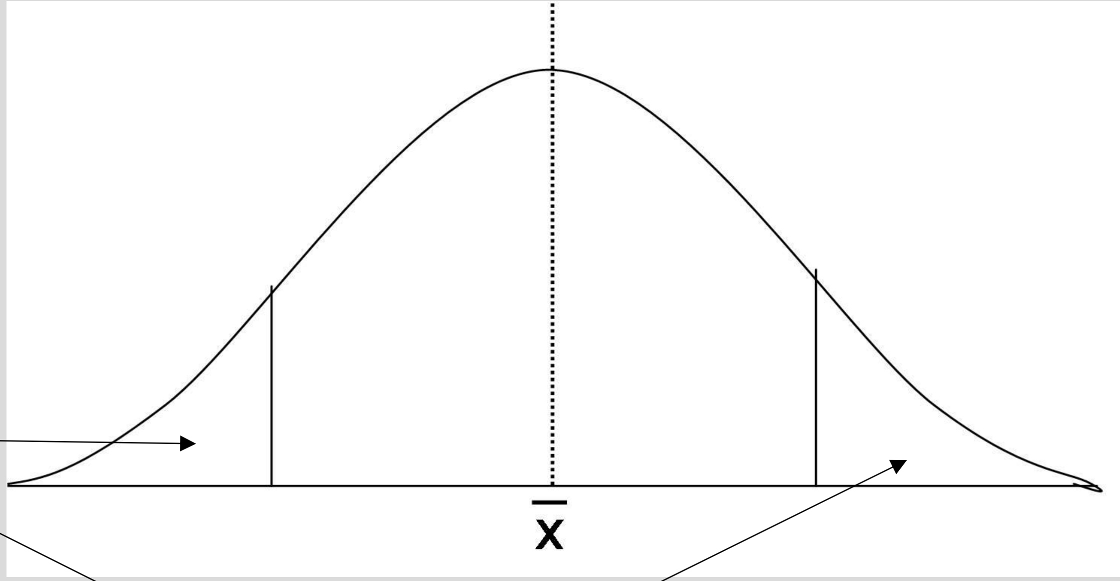
- Redial Dataset ----> Extended Redial & INSPIRED Dataset
- Recorded conversations between two humans SEEKER and RECOMMENDER talking about movies
  - Minimum conversation length
  - Minimum movies mentioned
  - Recommendation Requirements
    - Movie Description (purple)
    - Personal Opinion (green)
    - Wiki Info (pink)
    - Plot Description (peach)
    - Review (cyan)
- No turn order requirements

Sample Conversation:



Calculating Quality Factor: Length

- Defined as the number of words after removing stop words
- Length of explanations may influence how users perceive the explanation quality.
- Find mean and standard deviation for length in the dataset
- Calculate z score of a conversation.
- If z score is 2.5 deviations away, score 0
- If z score is negative, apply penalty
- If z score is positive apply smaller penalty



Conversation too long / short, score 0

Calculating Quality Factor: Readability

- How easy a conversation is to read.
  - Determined by number of words in a sentence
  - Number of syllables per word.
- Flesch Kincaid Reading Ease score
  - Reading levels (1st grade, 7th grade etc)
  - 8th grade is an average value
  - Higher values represent an easier read

```
score = 206.835 -(1.015 * (totalWords/totalSentences))
                -(84.6 * (totalSyllables/totalWords))
```

Calculating Quality Factor: Word Importance

- The sum of how impactful each word in a conversation is.
- Term-Frequency Inverse Document Frequency

$$\mathrm{TF}(t,d) = \frac{\text{Number of times term } t \text{ appears in document } d}{\text{Total number of terms in document } d}$$

$$\mathrm{IDF}(t,D) = \log\left(\frac{\text{Total number of documents in the corpus } N}{\text{Number of documents containing term } t}\right)$$

$$\text{TF-IDF}(t,d,D) = \mathrm{TF}(t,d) \times \mathrm{IDF}(t,D)$$

Calculating Quality Factor: Repetition

- How many duplicate words are in a conversation after stop words have been removed

```
#Repitition functions
def scoreRepitition(idList, wholeConv):
    repitionScores = []
    #Loop over conversations
    for id in idList:
        repeatedWords = 0
        curString = " ".join(wholeConv[hash(id)][0])

        #remove stop words
        tokenizedString = nltk.word_tokenize(curString)
        setString = set(tokenizedString)
        if STOP_WORDS.intersection(setString):
            setString -= STOP_WORDS

        #Search for repeated words
        for word in setString:
            if tokenizedString.count(word) > 1:
                repeatedWords +=1
        repitionScores.append(repeatedWords)
    return repitionScores

#End repitition function
```

Calculating Quality Factor: Subjectivity & Polarity

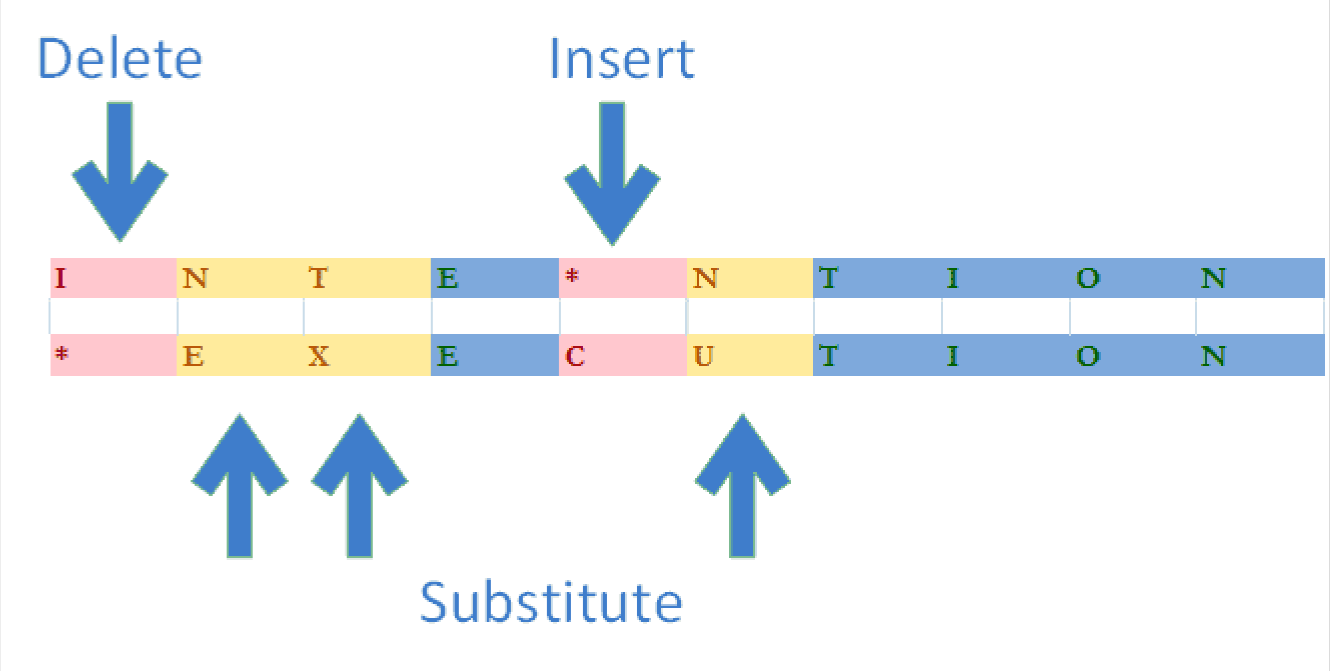
- Subjectivity measures how much a conversation contains personal opinion, emotion, and/or judgement.
- Polarity measures if the tone of a conversation is positive negative or neutral.
- Calculated by using the TextBlob python Library.

```
curString = " ".join(wholeConv[hash(id)][0])
blob = TextBlob(curString)
subjectivityScores.append(blob.sentiment.subjectivity)
```

```
curString = " ".join(wholeConv[hash(id)][0])
blob = TextBlob(curString)
polarityScores.append(blob.sentiment.polarity)
```

Calculating Quality Factor: Grammar

- Number of spelling errors after stop words and punctuation has been removed (Ignores movie titles*)
- Python spellchecker library.
  - Modified Levenshtein distance



Calculating Quality Factor: Feature Appearance

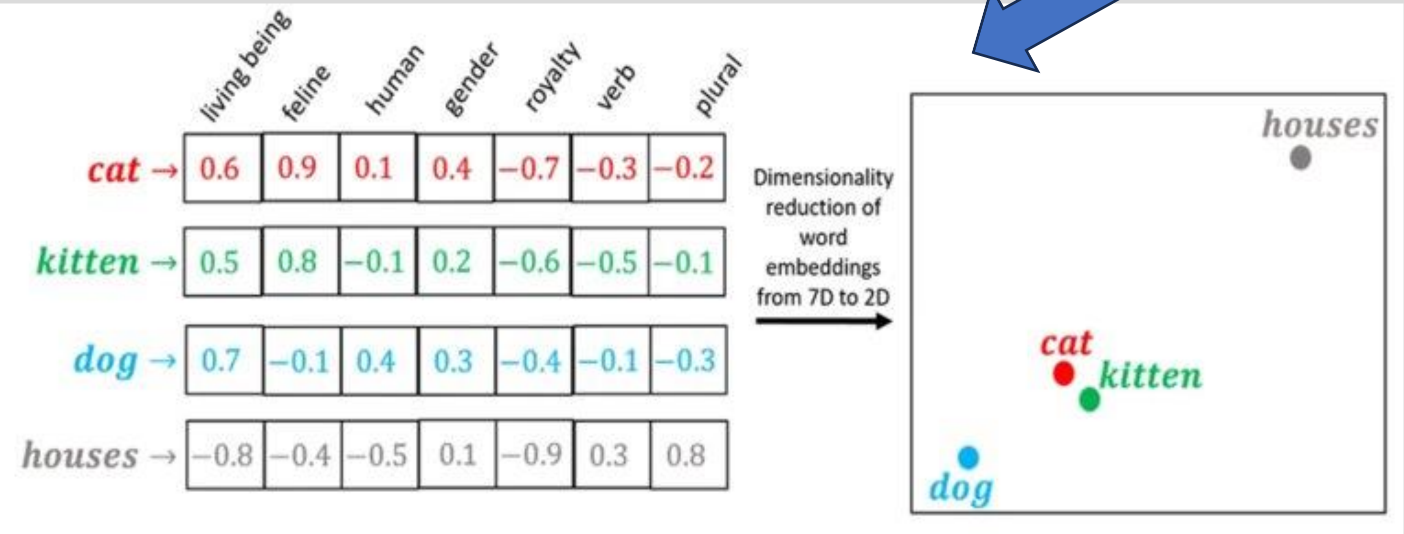
1. Divide conversation into 2 parts

Seeker | Recommender

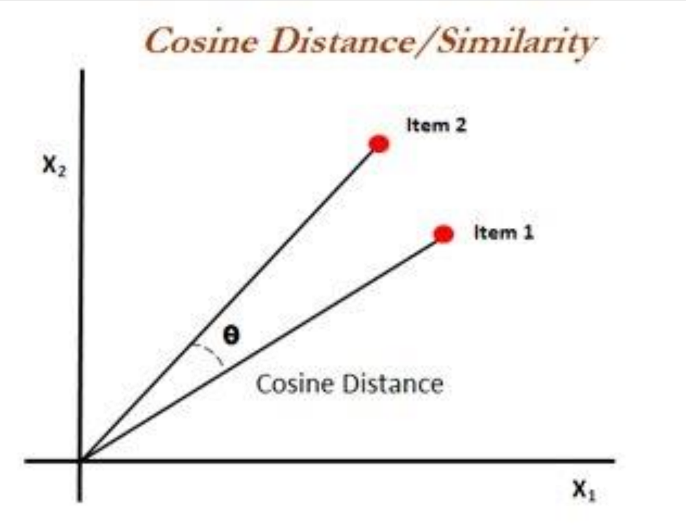
2. Use BART to summarize each half

Seeker Summary | Recommender Summary

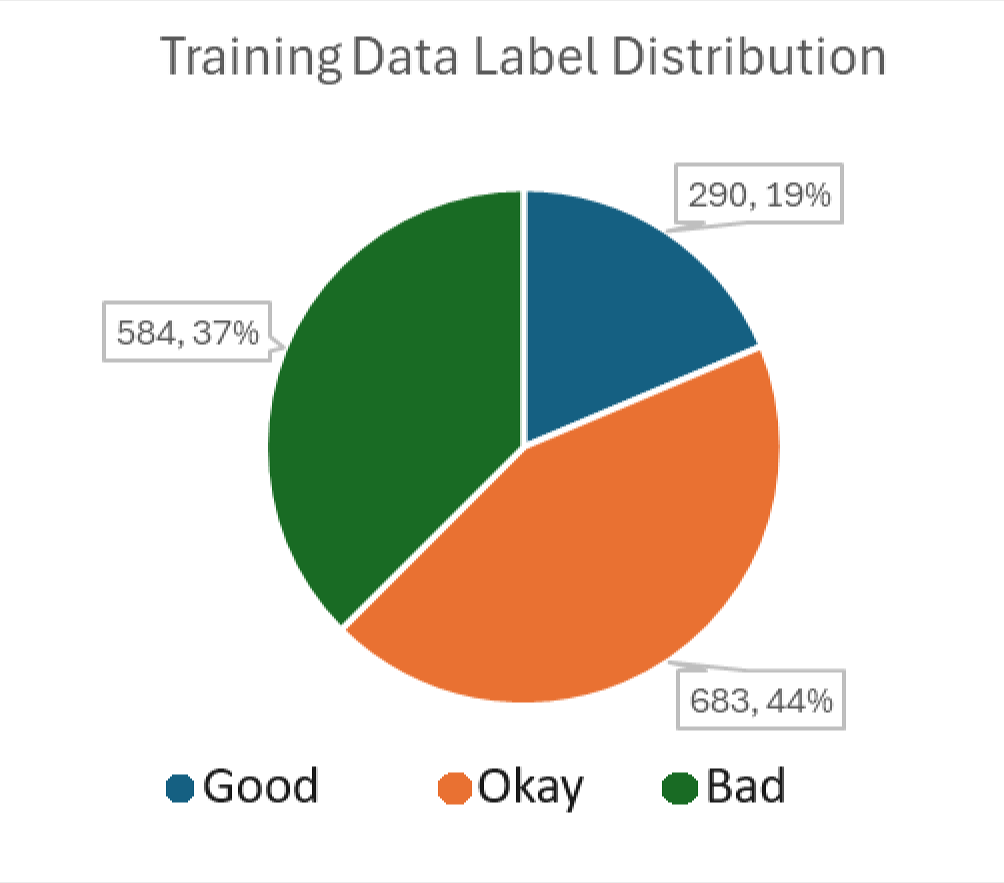
3. Embed Summaries with BERT

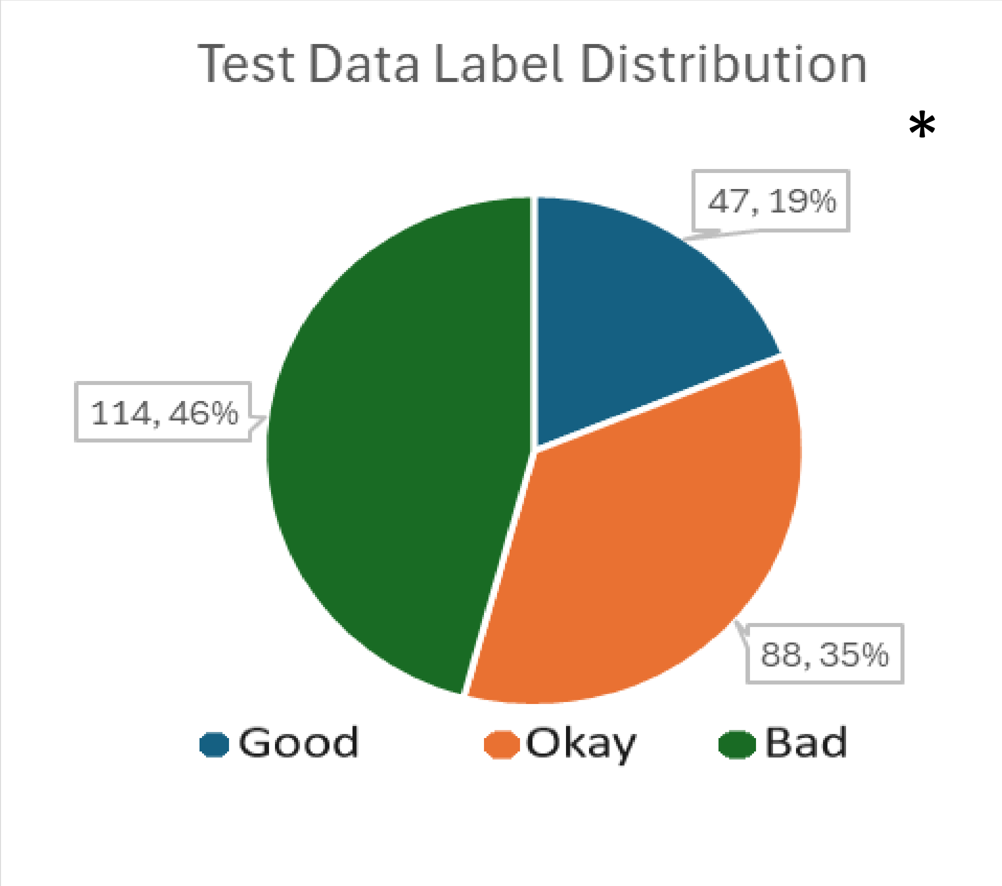


4. Calculate Cosine Similarity of Summary Embeddings



Target Label Distribution:





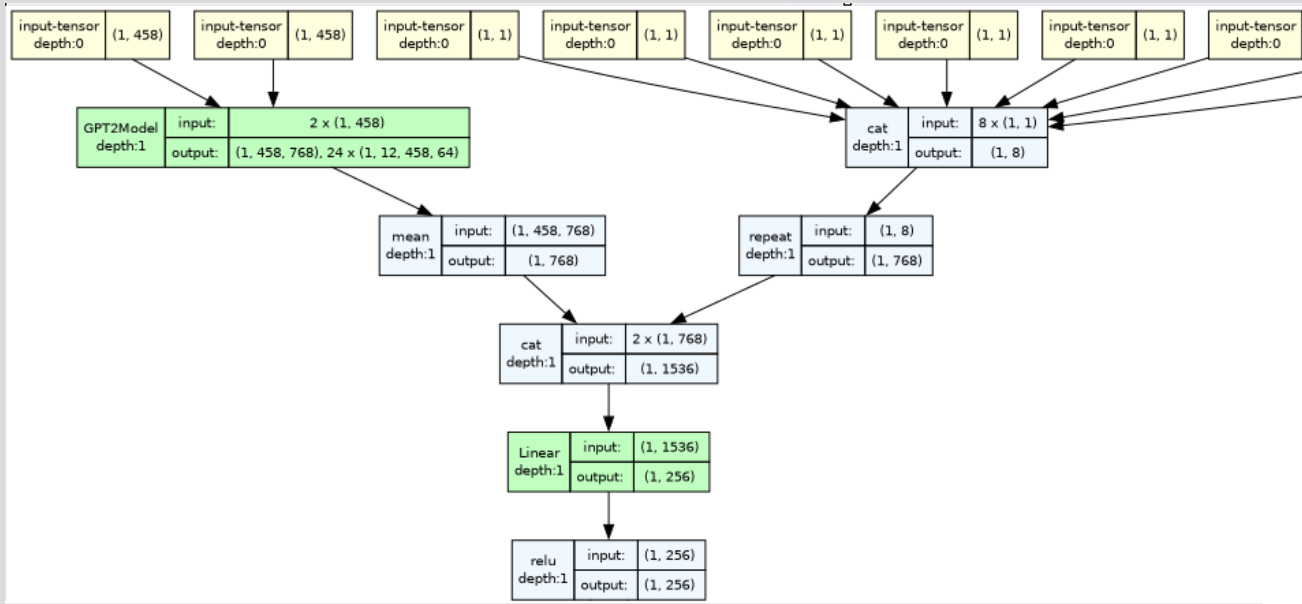
*Imbalanced test set explicitly part of the dataset. 823 of the system responses in the E-Redial test set are idle with no movie recommendations

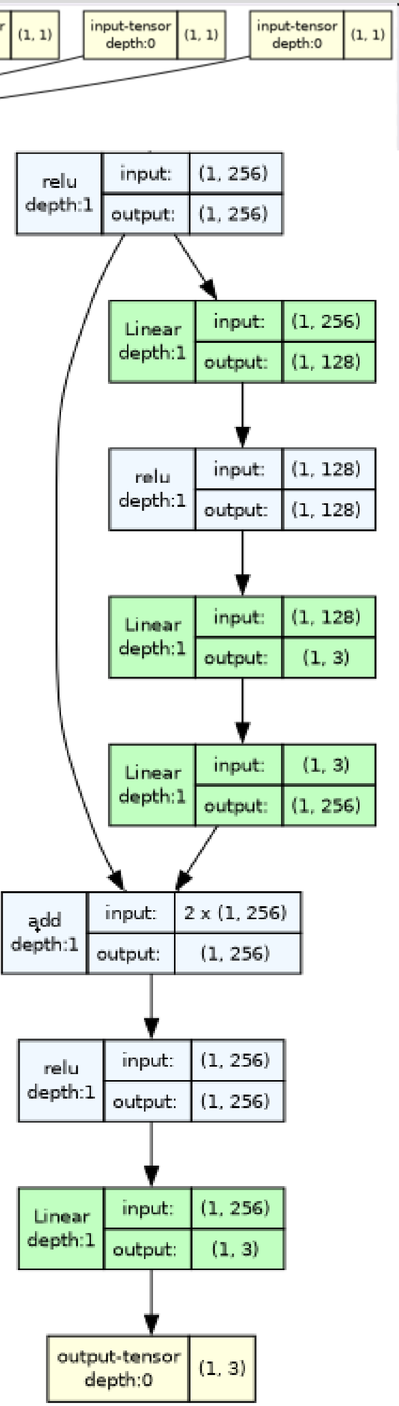
Model Architecture: Base Models – GPT2, GPT-NEO, T5

GPT2	NEO	T5
Transformer Architecture	Transformer Architecture	Transformer Architecture
Language Modelling Objective / token prediction	Language Modelling Objective / token prediction	Text-to-Text Objective
124 million parameters	125 million parameters	222 million parameters

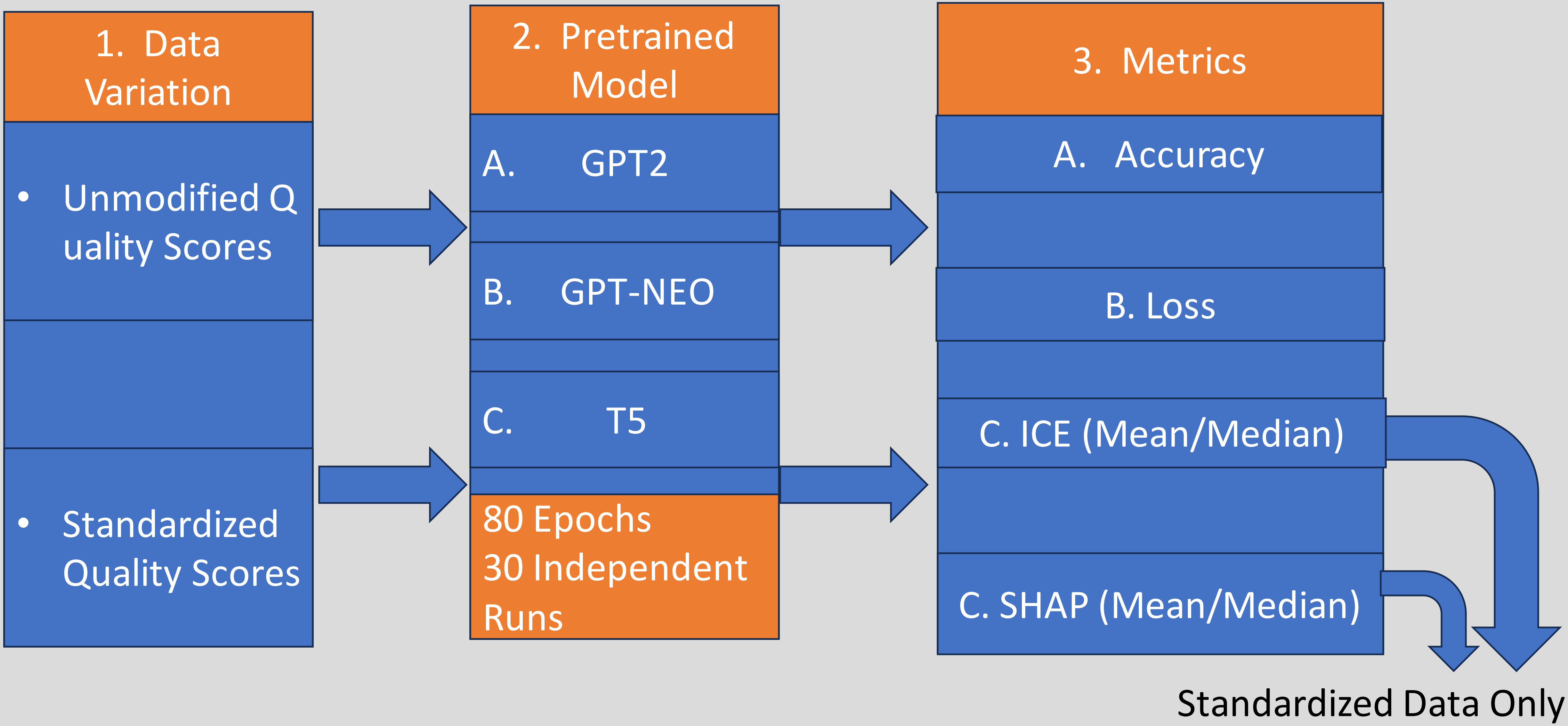
Model Architecture

- Input:
  - Embedded Conversation
  - 8 Quality Factors for the conversation
- Output:
  - Class Label {Good(0),Okay(1),Bad(2)}
- Architecture:
  - Base Model (GPT2, NEO, T5)
  - 3 blocks of 3 linear layers (256,128,3) with residual connections between the start and end of the block.

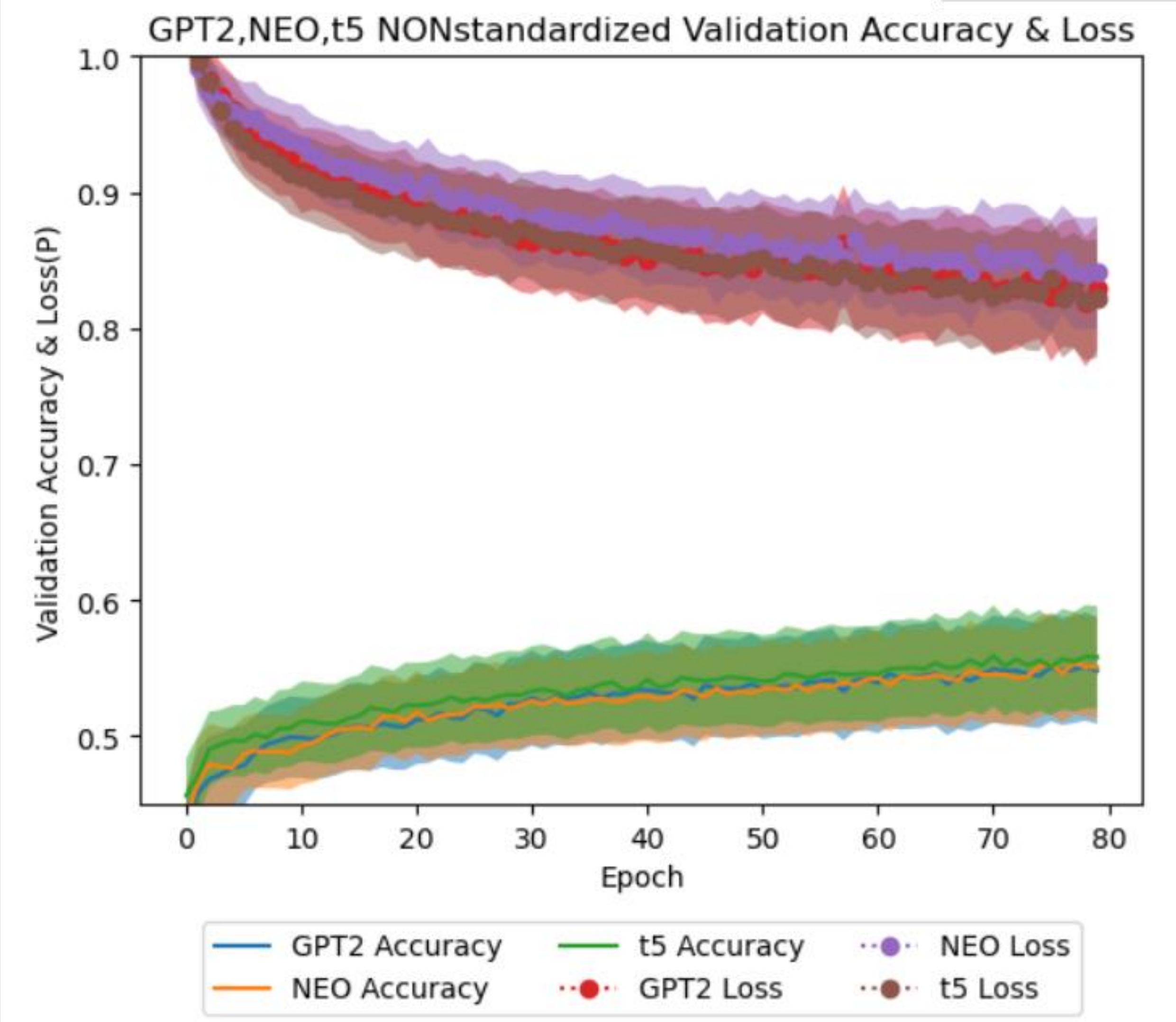


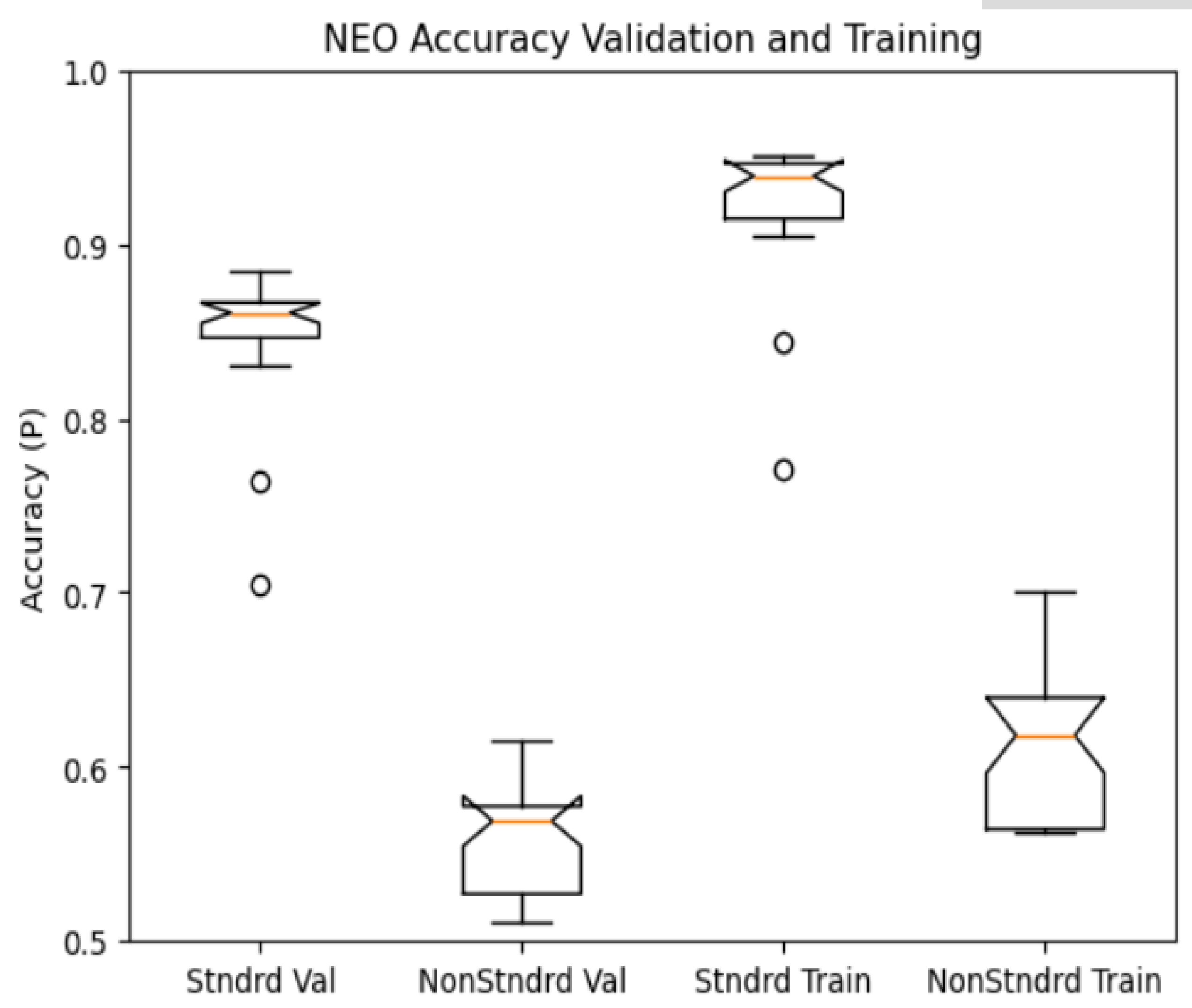


Experiment Details

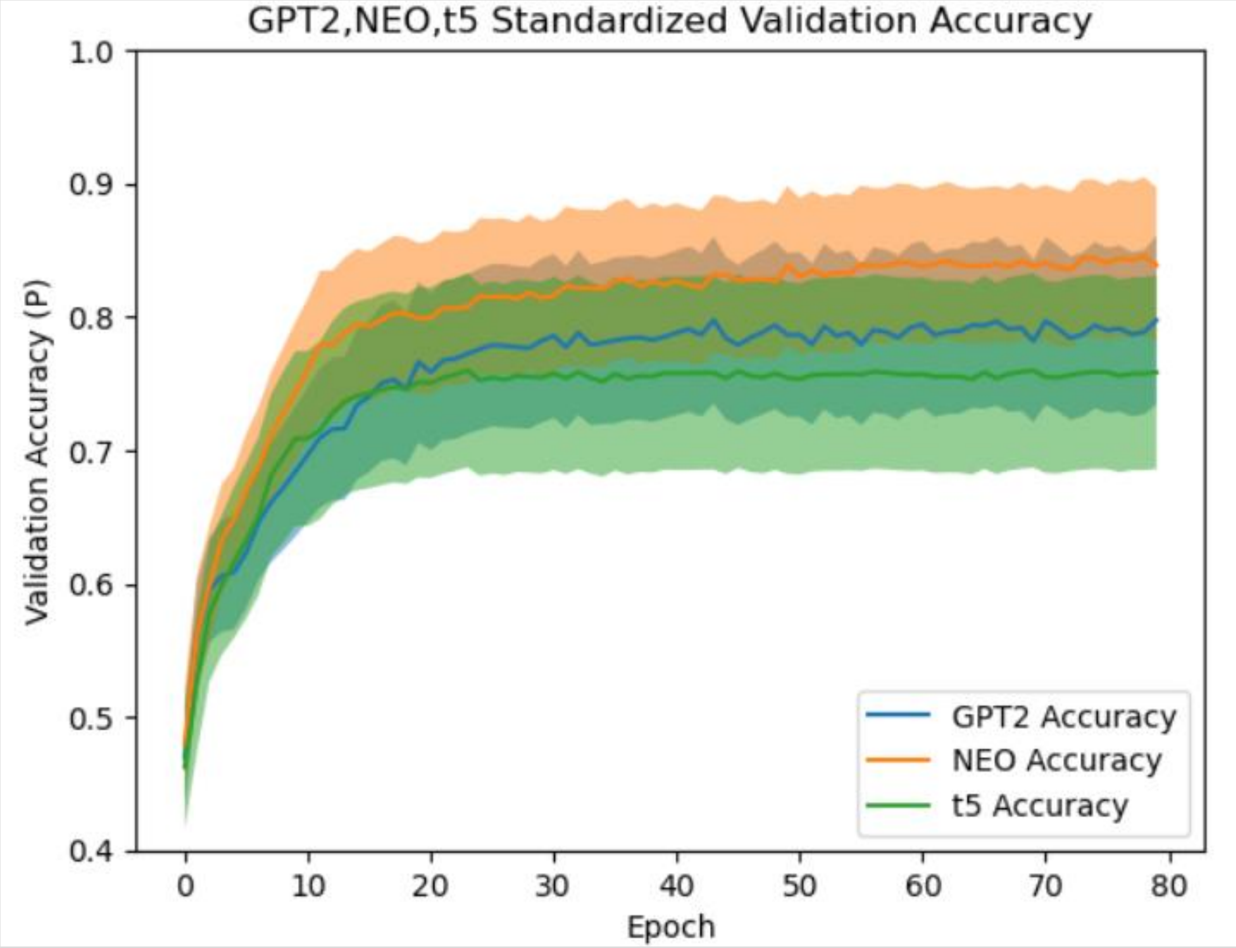


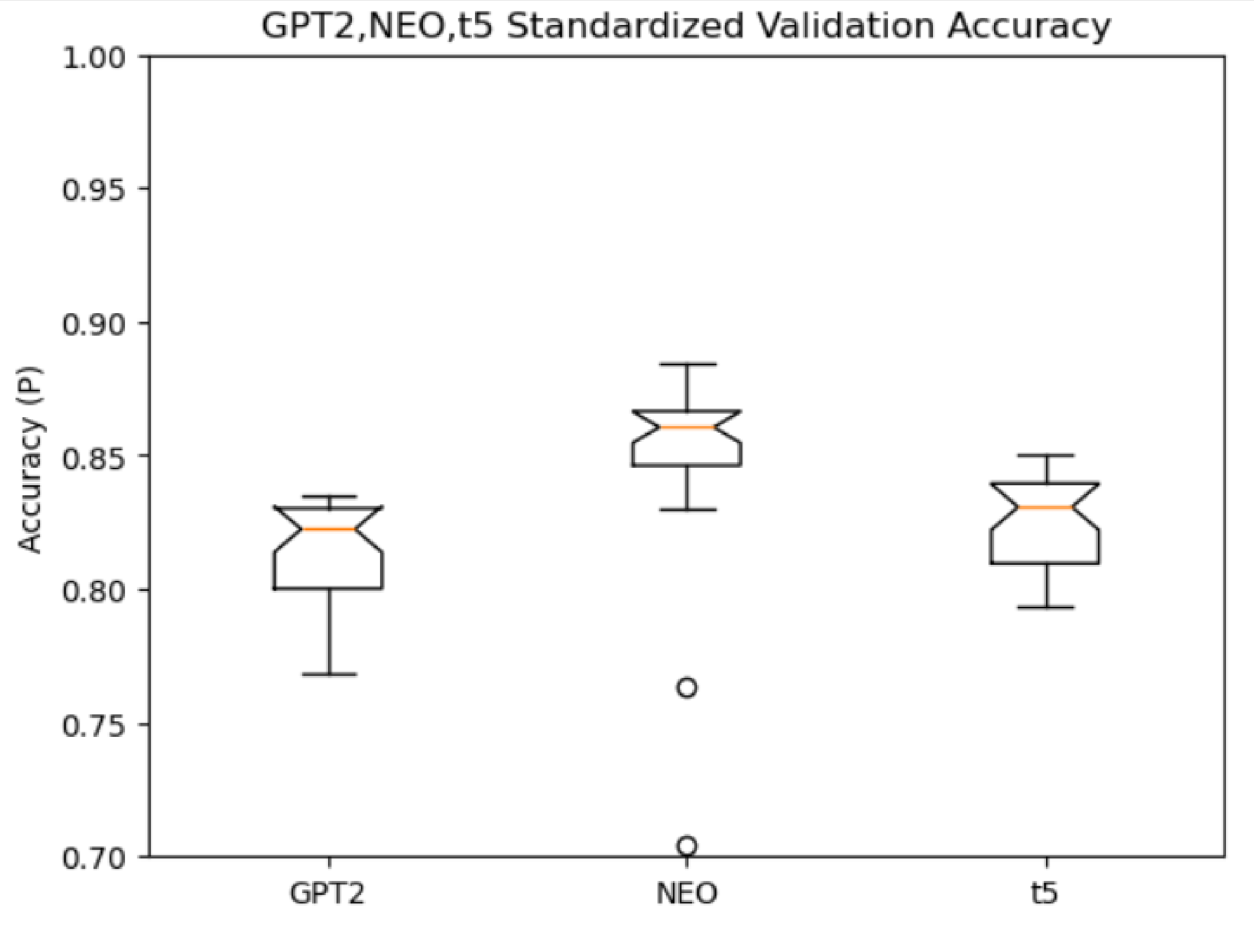
Results: NonStandardized Validation & Training





Results Standardized: Validation Set





Individual Conditional Expectation (ICE)

- Plot how model predictions change for individual instances as a single input feature changes (other inputs held constant)
- Useful for understanding the relationship between a feature and the model's predictions across different instances.
- Done in 5 steps:
  - Loop over 8 quality factors as QF
  - Loop over conversations in the validation set
  - Grab input data hold everything except QF constant
  - Vary QF value from 0.0-1.0
  - Predict class label, record results

Shapley Additive explanations (SHAP)

- SHAP values quantifying the contribution of each feature to the difference between the model's output for a given instance and the average model output.
- Help understand the relative importance of different features for that prediction.
- Done in 7 steps:
    - Establish background dataset (first 82 covnersations)
    - Loop over 8 quality factors as QF
    - Loop over conversations in the validation set
    - Grab input data hold everything except QF constant
    - Swap QF value with that same QF value from a different conversation in the background dataset
    - Predict class label,
    - Calculate SHAP value by taking the difference between the model prediction, on original data versus prediction on altered data, record results

ICE and SHAP, Why Both?

ICE analysis helps in understanding how predictions vary across instances as a single feature changes.

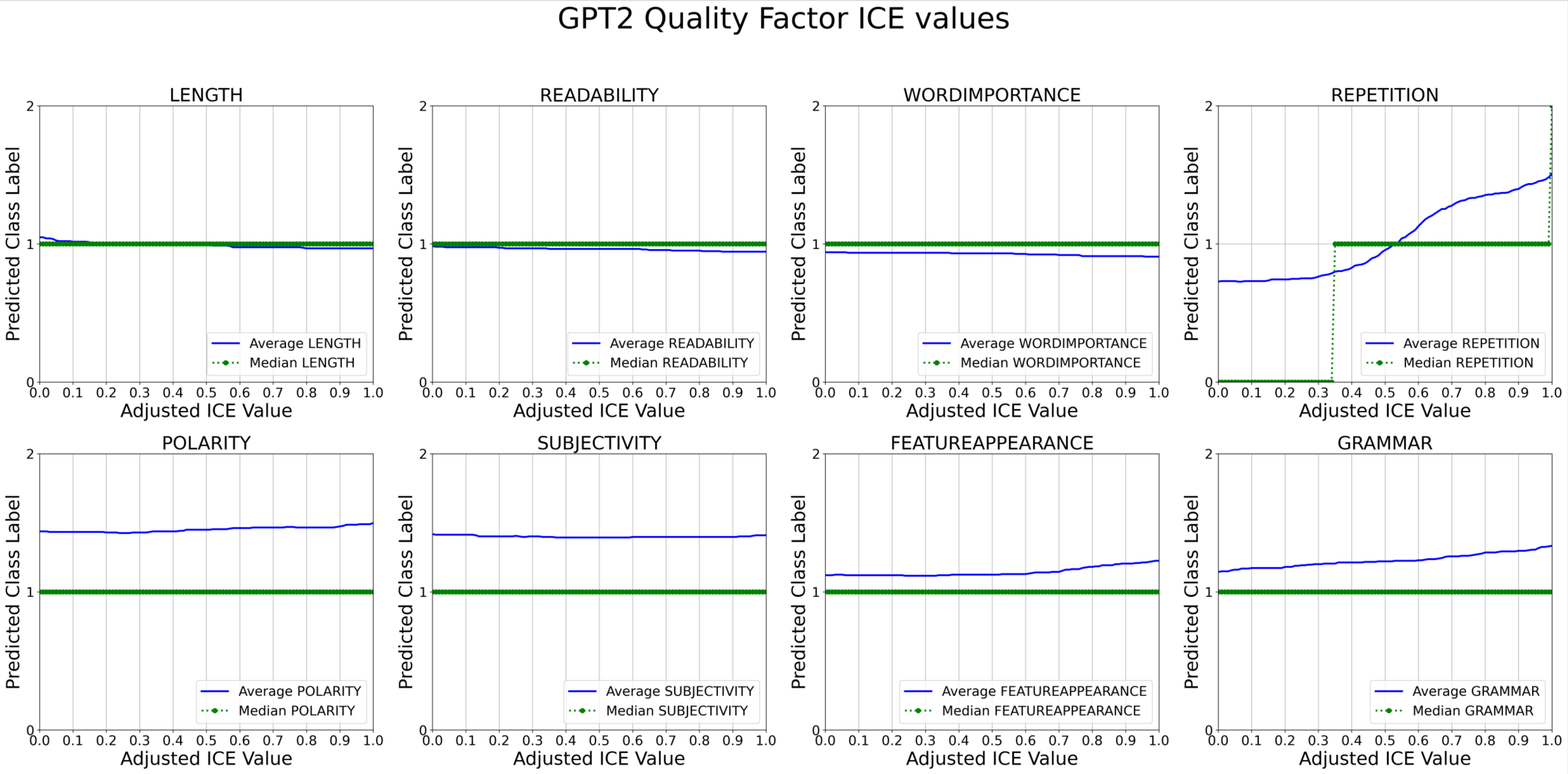
- Explains model behavior at the individual level
- Case by case explanation
- Exhaustive
- What if scenarios

SHAP analysis quantifyies the contribution of each feature to a prediction.

- Explains model behavior at a global level
- Big Picture explanation
- Smaller tweaks

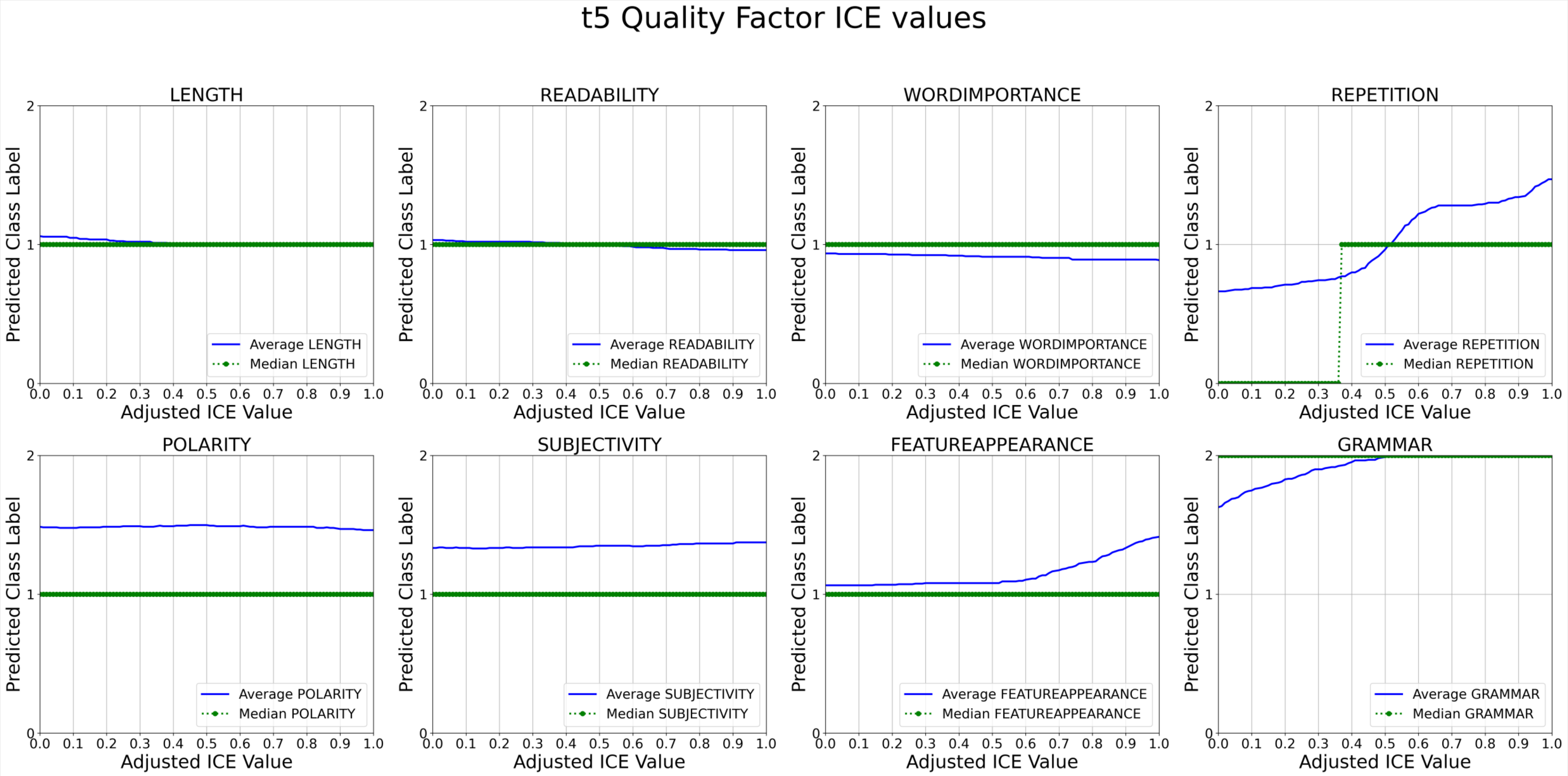
ICE Results: GPT2

Mean Median



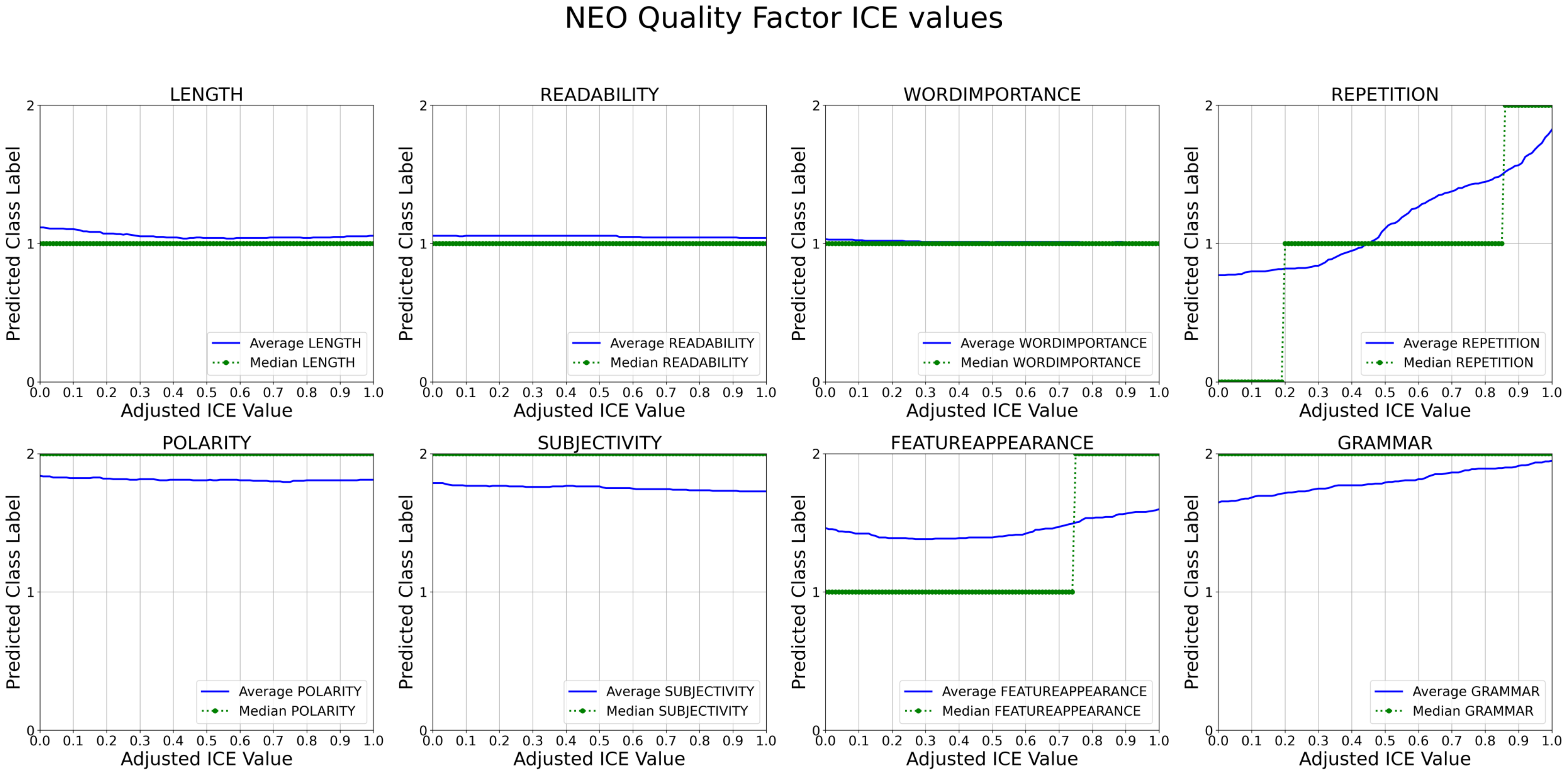
ICE Results: T5

Mean Median



ICE Results: NEO

Mean Median



ICE Results Summarized:

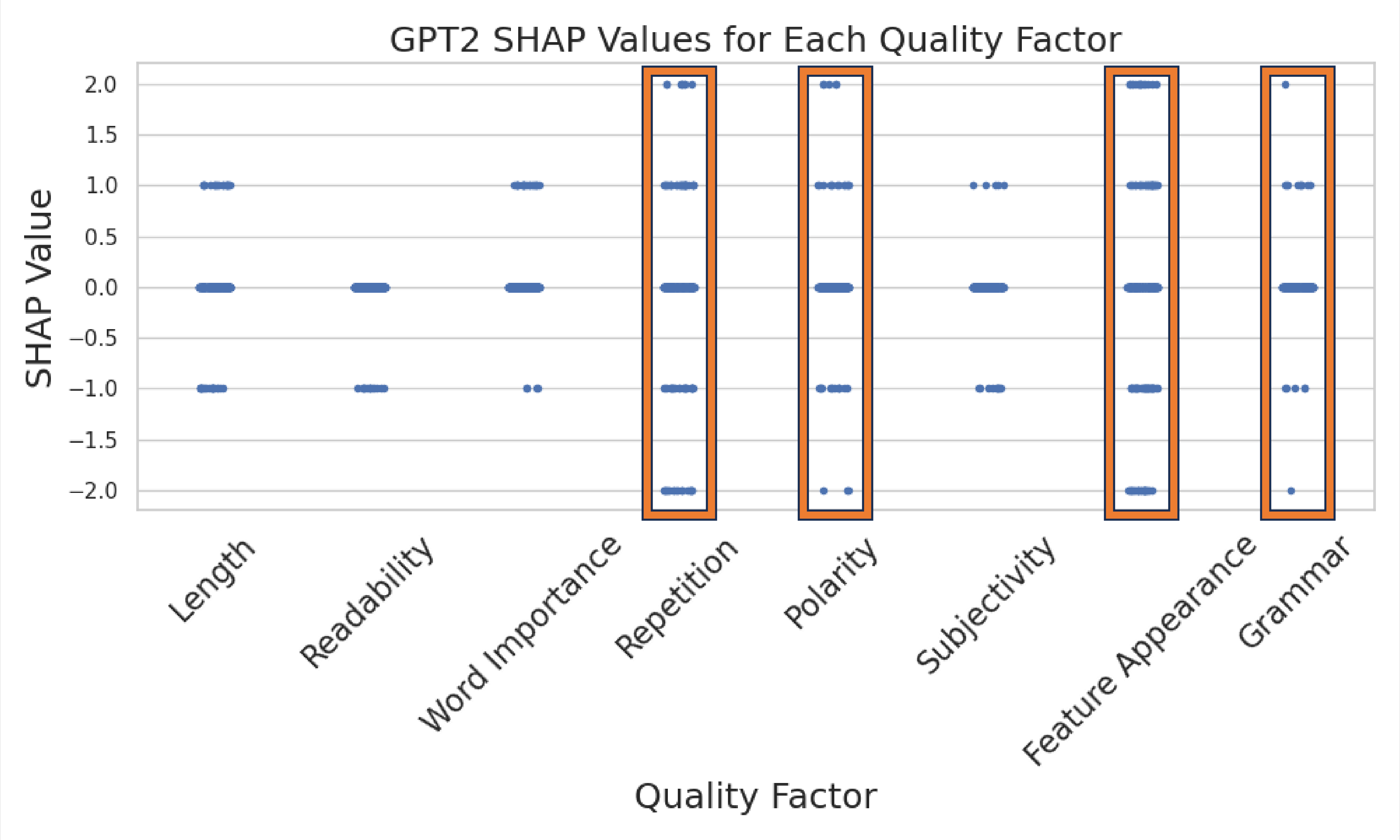
Individual input variations do not appear to alter classification much

The most impactful quality scores are:

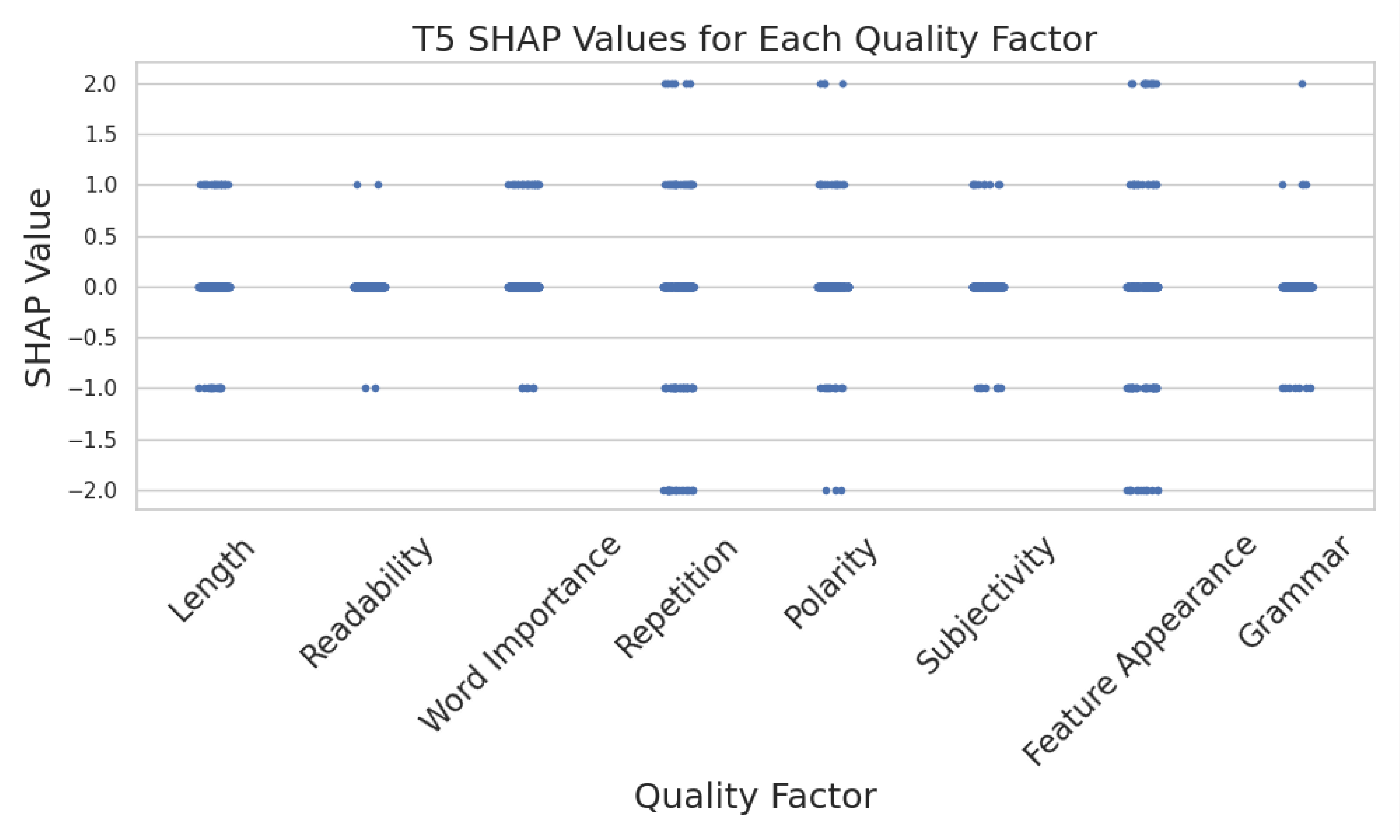
- Repetition
- Feature Appearance
- Grammar

The model predictions trend towards Okay class (1), which is the statistically safest bet.

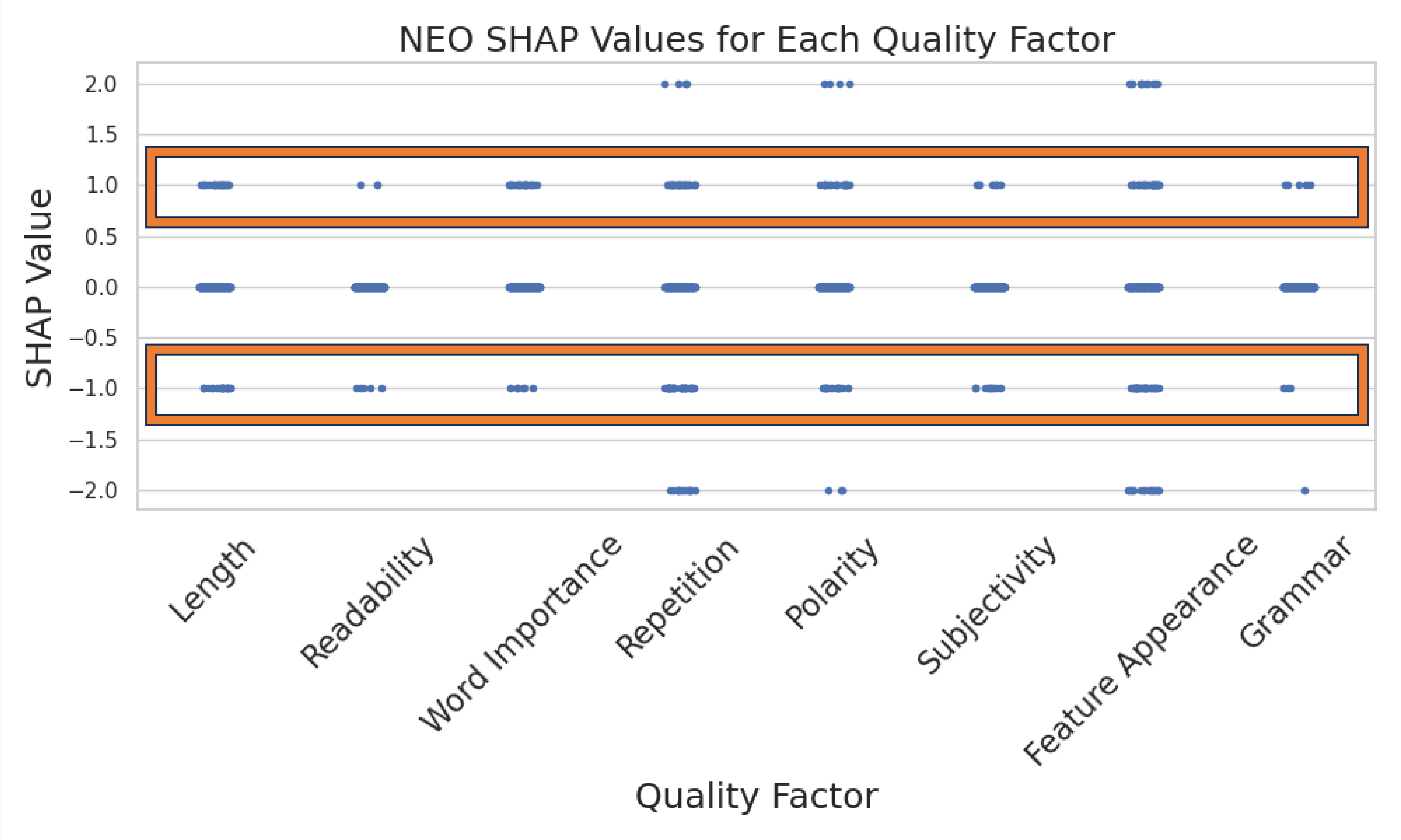
SHAP Results: GPT2



SHAP Results: T5



SHAP Results: NEO



SHAP Results Summarized

The strongest trend is for each QF value variation to not alter the predition (neutral effect)

Indicates individual factors are not as important as combinations of factors.

Repettion, Polarity, Feature Appearance, Grammar have the strongest effects on predictions, although each QF pushes predictions both positively and negatively.

NEO shows the most reactivity to alterations in QF values, GPT2 shows the highest variety of reactivity to QF alterations.

Overall Summary:

Goal: To find a set of evaluative metrics that accurately assess how well a recommendation request has been explained.

Combine 2 CRS datasets: E-Redial and INSPIRED

Score each conversation on 8 quality factors: Length, Readability, Repetition, Word Importance, Polarity, Subjectivity, Grammar, Feature Appearance

Based on Quality factors, assign each conversation a score {Good(0), Okay(1), Bad(2)}

Incorporate LLM / tranformer NLP base models to embed conversational data in conjunction with a residual network architecture

Discussion:

- On average, NEO is the best performing base model, GPT2 is second, and T5 performs worst on average
- Standardization of quality factor scores has a massive impact on model effectiveness
- Standardizing scores is the only way each score makes sense in context.
- Training results are better than validation results
- The model is most sensitive to alterations if 3 quality factors: Repetition, Feature Appearance, and Grammar
- Both ICE and SHAP show that individual QF changes have minimal to moderate effects
- Indicates that the model is using a combination of features rather than relying on a single QF

Discussion:

- Regardless of conversation type (SAUP, SAUE, etc) the 8 quality factors appear to be robust enough and useful for classifying conversational recommendations.
- All models had similar performances across ICE and SHAP analyses, and across training and validation sets.
- GPT2 and NEO had very similar behavior.
    - NEO is an open-source version of GPT2
    - NEO uses local attention in every other layer with a window size of 256 tokens.
    - Both models generate tokens sequentially based on previous input.
- T5 performs the worst
    - Architecture
    - Training data not as diverse

Questions?

Thanks!

References

[1] Banerjee, S. and Lavie, A. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In Goldstein, J., Lavie, A., Lin, C.-Y., and Voss, C., editors, Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization, pages 65–72. Association for Computational Linguistics.

[2] Barredo Arrieta, A., D́ıaz-Rodŕıguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., Garcia, S., Gil-Lopez, S., Molina, D., Benjamins, R., Chatila, R., and Herrera, F. Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. 58:82–115.

[3] Black, S., Biderman, S., Hallahan, E., Anthony, Q., Gao, L., Golding, L., He, H., Leahy, C., McDonell, K., Phang, J., Pieler, M., Prashanth, U. S., Purohit, S., Reynolds, L., Tow, J., Wang, B., and Weinbach, S. GPT-NeoX-20b: An open-source autoregressive language model.

[4] Caro-Mart́ınez, M., Jim´enez-D́ıaz, G., and Recio-Garćıa, J. A. Conceptual modeling of explainable recommender systems: An ontological formalization to guide their design and development. 71:557–589.

[5] Celikyilmaz, A., Clark, E., and Gao, J. Evaluation of Text Generation: A Survey, June

2020.

[6] Chen, X., Zhang, Y., and Wen, J.-R. Measuring ”why” in recommender systems: a comprehensive survey on the evaluation of explainable recommendation.

[7] Chen, Z., Wang, X., Xie, X., Parsana, M., Soni, A., Ao, X., and Chen, E. Towards explainable conversational recommendation. In Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI’20, pages 2994–3000.

[8] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding.

[9] Dietmar, J. and Ahtsham, M. End-to-End Learning for Conversational Recommendation: A Long Way to Go? pages 72–76, January 2020.

[10] Fayyaz, Z., Ebrahimian, M., Nawara, D., Ibrahim, A., and Kashef, R. Recommendation Systems: Algorithms, Challenges, Metrics, and Business Opportunities. Applied Sciences, 10(21):7748, January 2020.

References

[11] Finch, S. E. and Choi, J. D. Towards Unified Dialogue System Evaluation: A Comprehensive Analysis of Current Evaluation Protocols. In Proceedings of the 21th Annual Meeting of the Special Interest Group on Discourse and Dialogue, pages 236–245, 1st virtual meeting, July 2020. Association for Computational Linguistics.

[12] Flesch, R. A new readability yardstick. 32(3):221–233.

[13] Fu, Z., Xian, Y., Zhang, Y., and Zhang, Y. Tutorial on Conversational Recommendation Systems. In Proceedings of the 14th ACM Conference on Recommender Systems, RecSys '20, pages 751 753, New York, NY, USA, September 2020. Association for Computing Machinery.

[14] Gao, C., Lei, W., He, X., de Rijke, M., and Chua, T.-S. Advances and challenges inconversational recommender systems: A survey. AI Open, 2:100–126, January 2021.

[15] Goldstein, A., Kapelner, A., Bleich, J., and Pitkin, E. Peeking inside the black box: Visualizing statistical learning with plots of individual cond itional expectation.

[16] Guo, S., Zhang, S., Sun, W., Ren, P., Chen, Z., and Ren, Z. Towards explainable conversational recommender systems.

[17] Hayati, S. A., Kang, D., Zhu, Q., Shi, W., and Yu, Z. INSPIRED: Toward Sociable Recommendation Dialog Systems. In Procee dings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 8142–8152, Online, November 2020. Association for Computational Linguistics.

[18] Jannach, D. Evaluating Conversational Recommender Systems: A Landscape of Research. Artificial Intelligence Review, July 2022. arXiv :2208.12061 [cs].

[19] Kelly, D. Methods for Evaluating Interactive Information Retrieval Systems with Users. Foundations and Trends in Informa tion Retrieval, 3(1—2):1–224, January 2009.

[20] Krauth, K., Dean, S., Zhao, A., Guo, W., Curmei, M., Recht, B., and Jordan, M. I. Do Offline Metrics Predict Online Performance in Recommender Systems?, November 2020. arXiv:2011.07931 [cs].

References

[21] Lei, W., He, X., de Rijke, M., and Chua, T.-S. Conversational Recommendation: Formulation, Methods, and Evaluation. In Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '20, pages 2425–2428, New York, NY, USA, July 2020. Association for Computing Machinery.

[22] Lei, W., He, X., Miao, Y., Wu, Q., Hong, R., Kan, M.-Y., and Chua, T.-S. Estimation-Action-Reflection: Towards Deep Interaction Between Conversational and Recommender Systems. In Proceedings of the 13th International Conference on Web Search and Data Mining, WSDM '20, pages 304–312, New York, NY, USA, January 2020. Association for Computing Machinery.

[23] Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., and Zettlemoyer, L. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In Jurafsky, D., Chai, J., Schluter, N., and Tetreault, J., editors, Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 7871–7880. Association for Computational Linguistics

[24] Li, R., Kahou, S., Schulz, H., Michalski, V., Charlin, L., and Pal, C. Towards Deep Conversational Recommendations, March 2019. arXiv:1812.07617 [cs, stat].

[25] Lin, C.-Y. ROUGE: A package for automatic evaluation of summaries. In Text Summarization Branches Out, pages 74–81. Association for Computational Linguistics.

[26] Liu, C.-W., Lowe, R., Serban, I., Noseworthy, M., Charlin, L., and Pineau, J. How NOT To Evaluate Your Dialogue System: An Empirical Study of Unsupervised Evaluation Metrics for Dialogue Response Generation. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pages 2122–2132, Austin, Texas, November 2016. Association for Computational Linguistics.

[27] Lundberg, S. and Lee, S.-I. A unified approach to interpreting model predictions.

[28] Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. BLEU: a method for automatic evaluation of machine translation. In Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, ACL '02, pages 311–318. Association for Computational Linguistics.

[29] Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. Language models are unsupervised multitask learners.

[30] Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. Exploring the limits of transfer learning with a unified text-to-text transformer.

[31] Ramos, J. E. Using TF-IDF to determine word relevance in document queries.

References

[32] Sezerer, E. and Tekir, S. A survey on neural word embeddings.

[33] Sinha, K., Parthasarathi, P., Wang, J., Lowe, R., Hamilton, W. L., and Pineau, J. Learning an Unreferenced Metric for Online Dialogue Evaluation, May 2020. arXiv:2005.00583 [cs].

[34] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. Attention is all you need.

[35] Vultureanu-Albi ș i, A. and B ˘adic ˘a, C. Recommender systems: An explainable AI perspective. In 2021 International Conference on INnovations in Intelligent SysTems and Applications (INISTA), pages 1–6.

[36] Wen, B., Feng, Y., Zhang, Y., and Shah, C. ExpScore: Learning metrics for recommendation explanation. In Proceedings of the ACM Web Conference 2022, WWW ’22, pages 3740–3744. Association for Computing Machinery.

[37] Wong, C.-M., Feng, F., Zhang, W., Vong, C.-M., Chen, H., Zhang, Y., He, P., Chen, H., Zhao, K., and Chen, H. Improving Conversational Recommender System by Pretraining Billion-scale Knowledge Graph. In 2021 IEEE 37th International Conference on Data Engineering (ICDE), pages 2607–2612, April 2021. ISSN: 2375-026X.

[38] Zhang, Y., Chen, X., Ai, Q., Yang, L., and Croft, W. B. Towards Conversational Search and Recommendation: System Ask, User Respond. In Proceedings of the 27th ACM International Conference on Information and Knowledge Management, pages 177–186, Torino Italy, October 2018. ACM.

[39] Zhou, K., Wang, X., Zhou, Y., Shang, C., Cheng, Y., Zhao, W. X., Li, Y., and Wen, J.-R. CRSLab: An Open-Source Toolkit for Building Conversational Recommender System. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations, pages 185–193, Online, August 2021. Association for Computational Linguistics.

[40] Zhou, K., Zhao, W. X., Bian, S., Zhou, Y., Wen, J.-R., and Yu, J. Improving Conversational Recommender Systems via Knowledge Graph based Semantic Fusion. In Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD ’20, pages 1006–1014, New York, NY, USA, August 2020. Association for Computing Machinery.

[41] Zhou, K., Zhao, W. X., Wang, H., Wang, S., Zhang, F., Wang, Z., and Wen, J.-R. Leveraging Historical Interaction Data for Improving Conversational Recommender System. In Proceedings of the 29th ACM International Conference on Information & Knowledge Management, CIKM ’20, pages 2349–2352, New York, NY, USA, October 2020. Association for Computing Machinery.