Sentence Salience Classification Using Linguistic and Semantic Features in Question-Answering Setting

Project Report

Madduru Sai Chandra Nikhil - 2024901010

December 2025

https://github.com/MadduruNikhil-IIITH/Squad-Salience-Sentence-Detection

Abstract

This project develops and evaluates a fully interpretable sentence salience classifier for extractive question answering using only hand-crafted linguistic and semantic features. From the official SQuAD v1.1 **training set** (total 19,035 available passages), we randomly sampled 2,000 passages, resulting in 8,328 sentences. Each sentence was labeled as "Answer" (class 1) if it contains any part of a gold answer span. A total of 26 features — including surface, lexical, POS, discourse, and token-level surprisal from GPT-2 and BERT — were extracted. A Logistic Regression classifier with balanced class weights achieved **69.27% accuracy** and **0.7377 F1-score** on the positive (Answer) class. Feature importance analysis confirms that `sentence_position` and GPT-2 surprisal metrics are by far the strongest predictors.

1 Introduction and Research Question

While end-to-end neural models dominate modern QA, lightweight and explainable sentence-level classifiers remain useful for pipeline systems, retrieval augmentation, and low-resource deployment. This work answers:

Can non-neural linguistic and semantic features, particularly token-level surprisal from pretrained language models, reliably predict sentence salience (i.e., whether a sentence contains part of the answer) in the SQuAD dataset?

2 Dataset

- **Source**: SQuAD v1.1 `train-v1.1.json` (official training split)
- **Total passages available**: 19,035
- **Sampled passages**: 2,000 (random seed 2024901010)
- **Sentence segmentation**: NLTK Punkt
- **Labeling**: Binary — 1 if sentence overlaps any gold answer span

Statistic (2,000 passages)	**Value**
Total sentences	8,328
Answer sentences (class 1)	5,178
Non-answer sentences (class 0)	3,150
Answer ratio	62.17 %
Average sentences per paragraph	4.18
Average sentence length (words)	27.01

Table 1: Dataset statistics from the 2,000-passage training-set sample.

3 Feature Engineering

26 features were extracted per sentence (see Table 2):

Category	**Features**
Surface	`sentence_length_words`, `sentence_position`, `sentence_position_norm`
Lexical	`type_token_ratio`, `lexical_density`
POS	`noun_ratio`, `verb_ratio`, `adj_ratio`, `pronoun_ratio`
Discourse	`named_entity_density`, causal/contrast marker ratios
Surprisal (GPT-2)	mean, sum, std, var, min, max (token-level)
Surprisal (BERT)	mean, sum, std, var, min, max (masked token)

Table 2: Full feature set.

Surprisal was computed on GPU using `gpt2` (autoregressive) and `bert-base-uncased` (masked LM).

4 Methodology

- **Model**: Logistic Regression (`C=1.0`, `class_weight='balanced'`, max_iter=1000)
- **Preprocessing**: `StandardScaler`
- **Split**: 80/20 stratified train/test within the 2,000 passages
- **Evaluation**: Accuracy and F1-score (positive class = Answer)

5 Results

Model	Accuracy	F1	Precision	Recall
All 26 features	**69.27%**	**0.7377**	**0.7860**	**0.6950**
Top-10 features only	67.41%	0.7239	0.7648	0.6873
No-Surprisals	69.21%	**0.7416**	0.7756	**0.7104**

Table 3: Main result and ablation (2,000 passages from training set).

Passages	Sentences	Answer Ratio	Accuracy	F1
100	577	73.5%	70.7%	0.793
250	1,236	66.6%	63.3%	0.709
500	2,277	68.3%	66.0%	0.740
1,000	4,485	63.8%	69.1%	0.748
2,000	**8,328**	**62.2%**	**69.3%**	**0.738**

Table 4: Scaling behavior across training-set sample sizes.

Passages	Full Model	No Surprisal	Top-10
100	70.69 / 79.27	67.24 / 75.95	66.38 / 76.07
250	63.31 / 70.93	**67.74 / 75.16**	64.11 / 71.20
500	66.01 / 74.04	65.57 / 73.43	65.57 / 73.61
1,000	69.12 / 74.80	67.00 / 72.79	66.78 / 72.76
2,000	69.27 / 73.77	**69.21 / 74.16**	67.41 / 72.39

Table 5: Ablation study (Accuracy / F1 %). Bold = best per row.

Rank	Feature	Coefficient
1	`sentence_position`	−0.796
2	`gpt2_surprisal_sum`	+0.663
3	`gpt2_surprisal_var`	−0.444
4	`bert_surprisal_std`	+0.364
5	`noun_ratio`	+0.331
6	`bert_surprisal_var`	−0.315
7	`named_entity_density`	+0.298
8	`gpt2_surprisal_std`	+0.283
9	`sentence_position_norm`	+0.257
10	`lexical_density`	−0.256

Table 6: Top 10 features by absolute coefficient (2,000-passage model).

6 Discussion and Insights

- `sentence_position` is the dominant signal — answers overwhelmingly appear in the first few sentences of SQuAD paragraphs.
- GPT-2 surprisal features consistently outperform BERT surprisal, indicating that forward predictability is more relevant than bidirectional context for salience.
- High noun ratio and named-entity density strongly favor answer sentences.
- The model achieves strong performance using only lightweight, fully interpretable features — no passage encoder or BERT embeddings required.
- **Counter-intuitive ablation result**: Despite large positive coefficients for GPT-2 features (e.g., `gpt2_surprisal_sum` = +0.663), removing all surprisal features improved F1 from 73.77% to **74.16%** on 2,000 passages — indicating multicollinearity and noise in linear models.

7 Conclusion

Using only 2,000 passages from the SQuAD v1.1 **training set** and hand-crafted features, we built an interpretable sentence salience classifier reaching 69.3% accuracy and 0.738 F1 (answer class). The system demonstrates that classic linguistic cues combined with modern surprisal estimates from pretrained LMs capture a substantial portion of the answer-location signal, offering a fast, explainable alternative or complement to neural QA pipelines.

References

[1] Rajpurkar, P., Zhang, J., Lopyrev, K., & Liang, P. (2016). SQuAD: 100,000+ Questions for Machine Comprehension of Text. *arXiv:1606.05250.*

[2] Shen, D., & Klakow, D. (2006). Exploring Correlation of Dependency Relation Paths for Answer Extraction. *Proceedings of COLING-ACL 2006*, 889–896.

Appendix – Implementation

All code and results are available in the repository:

- `main.py`, `feature_extractor.py`, `surprisal.py` (GPU), `classifier.py`
- Results for all runs stored in `results/run_statistics.json`
- Final model and visualizations: `results/run_2000_passages/`
- https://github.com/MadduruNikhil-IIITH/Squad-Salience-Sentence-Detection