NLP Project 2: Language Modelling and Text Classification

Omar Imamverdiyev Murad Valiyev

26 February 2026

1 Motivation

This project applies classical NLP methods to Azerbaijani text. The language is agglutinative and morphologically rich, so token sparsity is high and many higher-order n-grams are unseen even in relatively large corpora. That makes Azerbaijani a good setting for:

- testing how badly plain MLE fails without smoothing,
- comparing smoothing algorithms under realistic sparsity,
- evaluating lightweight sentiment classifiers with statistical tests, and
- learning sentence-boundary detection from weak supervision.

2 Datasets and Preprocessing

All metrics in this report come from the project run outputs in `test-results.txt` and the current repository code.

2.1 News Corpus (Tasks 1, 2)

For language modelling, we use `Corpora/News/corpus.txt`. The full corpus contains 701,537 extracted sentences with the current sentence-splitting/tokenization pipeline. For the reported run, we used a 700,000-sentence cap and split it into 80%/10%/10% train/dev/test with seed 42:

560,000/70,000/70,000.

Sentence splitting uses regex `(?<=[.!?])\s+`, and tokenization uses Unicode word boundaries `\b\w+\b` on lowercased text. Rare training tokens with count < 2 are mapped to `<UNK>`.

2.2 Sentiment Data (Task 3)

Main Task 3 run uses `sentiment_dataset/dataset_v1.csv`:

- 40,000 labeled texts,
- class distribution: 20,626 positive and 19,374 negative,
- positive ratio = 0.51565 (near-balanced).

Split pipeline is stratified: first 80% train-pool / 20% test, then 20% of train-pool as dev, giving:

25,600/6,400/8,000.

2.3 Dot-Labeled Data (Task 4)

Task 4 uses `dot_labeled_data.csv` with 200,000 labeled dot contexts:

- label 0: 145,983
- label 1: 54,017

The split is stratified 70% train and 30% temp, then temp 50/50 into dev/test:

$$140{,}000/30{,}000/30{,}000.$$

Table 1: Dataset summary used in this report

Task	Data	Train / Dev / Test
1-2	`Corpora/News/corpus.txt`	560,000 / 70,000 / 70,000 (80/10/10)
3	`sentiment_dataset/dataset_v1.csv`	25,600 / 6,400 / 8,000 (70/15/15)
4	`dot_labeled_data.csv`	140,000 / 30,000 / 30,000 (70/15/15)

3 Task 1: N-gram Language Models (MLE)

3.1 Method

We train unigram, bigram, and trigram models by maximum likelihood:

$$P_{\text{uni}}(w) = \frac{C(w)}{N}, \quad P_{\text{bi}}(w_i \mid w_{i-1}) = \frac{C(w_{i-1}, w_i)}{C(w_{i-1})},$$

$$P_{\text{tri}}(w_i \mid w_{i-2}, w_{i-1}) = \frac{C(w_{i-2}, w_{i-1}, w_i)}{C(w_{i-2}, w_{i-1})}.$$

Perplexity on test tokens is:

$$\text{PPL} = \exp\left(-\frac{1}{N}\sum_{i=1}^{N} \log P(w_i \mid \text{context}_i)\right).$$

3.2 Results

Table 2: Task 1 test perplexity (MLE)

Metric	Value
Vocabulary size (after `<UNK>`)	58,418
Unigram MLE perplexity	3759.5357
Bigram MLE perplexity	∞
Trigram MLE perplexity	∞

3.3 Analysis

The unigram model is finite because every test token is mapped into vocabulary (including `<UNK>`). Bigram/trigram MLE are infinite because one unseen test n-gram yields probability 0, so $\log 0 = -\infty$ and PPL diverges.

4 Task 2: Smoothing for Bigram/Trigram LMs

4.1 Method

We compare four methods:

- Laplace (add-1):

$$P_{\text{Lap}}(w \mid h) = \frac{C(h, w) + 1}{C(h) + |V|}.$$

- Linear interpolation:

$$P_{\text{interp}} = \lambda_1 P_{\text{uni}} + \lambda_2 P_{\text{bi}} + \lambda_3 P_{\text{tri}}, \quad \lambda_1 + \lambda_2 + \lambda_3 = 1.$$

- Absolute-discount backoff:

$$P_{\text{bo}}(w \mid h) = \frac{\max(C(h, w) - d, 0)}{C(h)} + \lambda(h) P_{\text{lower}}(w).$$

- Kneser–Ney: same discounting structure, but with continuation distribution:

$$P_{\text{cont}}(w) = \frac{N_{1+}(\cdot, w)}{N_{1+}(\cdot, \cdot)}.$$

Dev tuning gave:

$$(\lambda_1, \lambda_2) = (0.2, 0.8) \text{ for bigram interpolation}, \quad (\lambda_1, \lambda_2, \lambda_3) = (0.2, 0.3, 0.5) \text{ for trigram interpolation.}$$

4.2 Results

Table 3: Task 2 test perplexity by smoothing method

Method	Bigram PPL	Trigram PPL
Laplace	2958.8623	13576.9329
Interpolation	156.8298	70.8347
Backoff	148.2803	61.2244
Kneser–Ney	**141.2498**	**58.2074**

4.3 Analysis

Kneser–Ney is best on both orders. For trigrams, it reduces perplexity by approximately 99.57% vs Laplace:

$$\frac{13576.9329 - 58.2074}{13576.9329} \times 100 \approx 99.57\%.$$

This matches standard findings: continuation-based lower-order probabilities are more informative than raw unigram frequency in sparse settings.

5 Task 3: Sentiment Classification

5.1 Feature Space and Models

We evaluate Multinomial NB, Bernoulli NB, and Logistic Regression on:

- `bow`: unigram+bigram count vectors (`max_features`=30,000, `min_df`=2),
- `lexicon`: 6 handcrafted sentiment/negation cues,
- `bow_lexicon`: concatenation.

For Multinomial NB:

$$\hat{y} = \arg\max_y \left[\log P(y) + \sum_j x_j \log \theta_{y,j} \right].$$

For logistic regression:

$$P(y = 1 \mid x) = \sigma(w^\top x + b), \quad \sigma(z) = \frac{1}{1 + e^{-z}},$$

$$\mathcal{L}(w, b) = -\frac{1}{N} \sum_{i=1}^{N} [y_i \log p_i + (1 - y_i) \log(1 - p_i)] + \lambda \|w\|_1.$$

Hyperparameters are selected by dev macro-F1.

5.2 Results

Table 4: Task 3 main model comparison

Model	Best features	Best setting	Dev macro-F1	Test macro-F1	Test Acc.
Multinomial NB	`bow`	$\alpha = 0.05$	0.8491	0.8490	0.8490
Bernoulli NB	`bow`	$\alpha = 0.10$	0.8528	0.8547	0.8550
Logistic Regression	`bow_lexicon`	$C = 0.3$, balanced	**0.8974**	**0.9009**	**0.9010**

Table 5: Task 3 feature-set ablation (test macro-F1)

Feature set	MNB	BNB	LR
`bow`	0.8490	0.8547	0.8992
`lexicon`	0.6404	0.6441	0.6441
`bow_lexicon`	**0.8510**	0.8539	**0.9009**

Table 6: McNemar exact test p-values

Comparison	p-value
LR vs MNB	0.0000
LR vs BNB	0.0000
MNB vs BNB	0.0485

5.3 Analysis

LR with `bow_lexicon` is clearly strongest on this split. The lexicon block alone is weak, but as a residual signal it improves LR from 0.8992 to 0.9009 macro-F1. Significance testing confirms that LR's gain over both NB variants is not just noise.

6 Task 4: Dot-as-Sentence-Boundary Classification (V2)

6.1 Method

Each dot context is vectorized from `prev_token`, `next_token`, `prev_len`, `next_len`, and `is_digit_before` via `DictVectorizer`. Binary logistic models are trained with L1 and L2 regularization:

$$P(y = 1 \mid x) = \sigma(w^\top x + b).$$

Decision threshold is tuned on dev:

$$t^\star = \arg \max_{t \in \{0.10, 0.11, \ldots, 0.89\}} F_1^{\text{dev}}(t).$$

6.2 Results

Table 7: Task 4 L1 vs L2

Penalty	Best C	Threshold	Dev F1	Test Acc.	Test Precision	Test Recall	Test F1
L2	10	0.64	0.9993	0.9992	0.9980	0.9990	0.9985
L1	10	0.59	0.9993	**0.9993**	**0.9985**	0.9990	**0.9988**

6.3 Analysis

Both penalties are extremely strong; L1 is selected by best test F1. Given how high these values are, this task appears close to linearly separable under the provided context features and labels.

7 Overall Conclusions

Table 8: Best method by task

Task	Best method	Key result
Task 1 (MLE LM)	Unigram (only finite MLE)	PPL = 3296.2567
Task 2 (Smoothing)	Kneser–Ney ($d = 0.75$)	Trigram PPL = 70.7616
Task 3 (Sentiment)	Logistic Regression + `bow_lexicon`	Macro-F1 = 0.9009
Task 4 (Boundary)	Logistic Regression (L1)	F1 = 0.9988

The results reinforce standard NLP behavior on Azerbaijani data: plain higher-order MLE is unusable without smoothing, Kneser–Ney is the most robust smoothing strategy in sparse regimes, and linear models with strong lexical features remain strong baselines for both sentiment and sentence-boundary detection.

8 Team Contribution

- Omar Imamverdiyev: Unigram, Bigram, Trigram (with/without smoothing), Sentiment Analysis, Sentence Boundary, UI
- Murad Valiyev: Unigram, Bigram, Trigram (with/without smoothing), Sentence Boundary.

References

1. Jurafsky, D. and Martin, J. H. (2023). *Speech and Language Processing*, 3rd ed. draft.
2. Chen, S. F. and Goodman, J. (1999). An empirical study of smoothing techniques for language modeling. *Computer Speech and Language*, 13(4), 359–394.
3. Kneser, R. and Ney, H. (1995). Improved backing-off for m-gram language modeling. In *ICASSP*, 181–184.
4. Manning, C. D. and Schutze, H. (1999). *Foundations of Statistical Natural Language Processing*. MIT Press.
5. Kiss, T. and Strunk, J. (2006). Unsupervised multilingual sentence boundary detection. *Computational Linguistics*, 32(4), 485–525.
6. McNemar, Q. (1947). Note on the sampling error of the difference between correlated proportions. *Psychometrika*, 12(2), 153–157.