Smoothing Techniques for Unigram Language Models: A Comparison of Lidstone and Held-Out Estimation

Julia Cher

December 2025

Abstract

Language models estimate probability distributions over words and are central to many NLP applications. A persistent challenge is data sparsity: events unseen in training receive zero probability under maximum likelihood estimation (MLE). Smoothing techniques address this limitation by redistributing probability mass to unseen events.

In this study, we implement and compare two classical smoothing methods - Lidstone's law and held-out estimation - within the framework of unigram language models. Using the Reuters-21578 corpus, the data is split into training, validation, and test sets. Model parameters are tuned on validation data, and performance is evaluated on test data using perplexity.

Our results show that MLE performs poorly due to zero-probability assignments. Lidstone smoothing with an optimized parameter achieves the lowest test perplexity, while held-out estimation provides a more flexible allocation of probability to rare and unseen events. These findings highlight the continued pedagogical and conceptual value of classical smoothing methods.

1 Introduction

Language models are a core component of natural language processing (NLP), supporting tasks such as speech recognition, machine translation, and information retrieval. These models estimate probability distributions over word sequences, providing a quantitative measure of how well a text conforms to the patterns of a language.

A central challenge in language modeling is data sparsity. Even large corpora contain many rare or unseen events, leading maximum likelihood estimation (MLE) to assign zero probability to words not observed in training. This limitation motivated the development of smoothing techniques, which redistribute probability mass to unseen events. Classical approaches such as Katz's back-off model [1] and the empirical comparison by Chen and Goodman [2] demonstrated the importance of smoothing for robust language modeling.

Among these techniques, Lidstone's law of succession and held-out estimation are two foundational methods. Lidstone smoothing adjusts observed counts by a constant parameter, while held-out estimation uses separate data to guide probability allocation for rare and unseen events. Although both methods are well studied, examining their behavior on real corpora provides insight into the mechanics of smoothing.

The objectives of this work are threefold: (1) to implement Lidstone and held-out unigram models using a subset of the Reuters-21578 corpus, (2) to evaluate their performance using perplexity, and (3) to compare how each method redistributes probability mass. We hypothesize that smoothing outperforms MLE, that tuned Lidstone smoothing yields lower perplexity than held-out estimation, and that held-out estimation assigns probabilities more flexibly to rare events.

Our experiments were implemented in Python using standard numerical and visualization libraries.[1]

[1] All source code and LaTeX files used to produce the results and figures in this paper are publicly available at: `https://github.com/juliacher/smoothing_lidstone_heldout_ulm.git`.

2 Methods

2.1 Dataset

Our experiments are based on datasets derived from the *Reuters-21578* corpus [3], a well-known benchmark in text categorization and language modeling research. The original collection consists of 21,578 articles in English newswire published by Reuters in 1987. Each document contains a header with topical labels and the body of the article. For this study, we used derived subsets of the corpus that have been formatted for language modeling tasks.

The data is divided into two files: a *development set* containing 4,248 articles and a *test set* containing 1,866 articles, for a total of 6,114 documents. Each article is represented by two lines, the first being a header and the second the article body. Since the focus of our research is word probability estimation rather than topic classification, only the body text is used for training and evaluation, while the header information is ignored. A summary of the statistics of the data set is presented in Table 1.

We applied preprocessing to the corpus prior to use. We tokenized the text in whitespace, converted all tokens to lowercase, and removed isolated punctuation symbols from the set `!  # , - .  : ; ?  @ ~`. We retained proper names, numbers, and other symbols as independent tokens to preserve the statistical properties of the text.

In addition to the dataset files, our program accepts an `INPUT WORD` as a parameter. This word may be present in the training data (a seen event) or absent from it (an unseen event). Including this parameter allows us to demonstrate how different smoothing methods assign probabilities to both observed and unobserved events.

We also assume a fixed vocabulary size of $|V| = 300{,}000$. This value does not correspond to the number of distinct tokens actually present in the dataset but represents an upper bound on the space of possible word events. By adopting this assumption, we ensure that the smoothing methods can distribute the probability mass not only across observed words, but also across the large set of potential unseen words.

Table 1: Summary statistics of articles in the derived Reuters-21578 datasets.

Property	Value
Number of articles (development set)	4,248
Number of articles (test set)	1,866
Total articles used	6,114
Original Reuters-21578 corpus size	21,578
Number of topics in original corpus	9
Assumed vocabulary size $\|V\|$	300,000
Tokenization	Whitespace, lowercase, punctuation removed

2.2 Unigram Model

We model the dataset as a sequence of independent word events drawn from a fixed vocabulary. For the purposes of this study, we assume a vocabulary size of $|V| = 300{,}000$. This vocabulary provides the basis for estimating probabilities with maximum likelihood and with smoothing methods.

The baseline unigram model is estimated using the Maximum Likelihood Estimate (MLE), where the probability of a word w is given by

$$P_{\text{MLE}}(w) = \frac{c(w)}{N},$$

with $c(w)$ the count of w in the training corpus and N the total number of tokens. Although simple, this approach assigns zero probability to unseen words, which limits its generalization.

2.3 Smoothing Methods

To address the zero-probability problem of the MLE unigram model, we applied two classical smoothing techniques: Lidstone's law and held-out estimation. Both methods redistribute probability mass from frequent events to rare or unseen ones, but they differ in how the allocation is determined.

Lidstone smoothing. In Lidstone smoothing each count is adjusted by the constant $\lambda > 0$, so that

$$P_{\text{Lid}}(w) = \frac{c(w) + \lambda}{N + \lambda|V|},$$

where $c(w)$ is the frequency of w in the training data, N is the total number of training tokens, and $|V|$ is the vocabulary size.

In our implementation, the development set was split into a *training portion* (90%) and a *validation portion* (10%). Probabilities were estimated from the training portion, while model performance was measured on the validation portion. This separation ensured that the smoothing parameter λ was tuned fairly.

To select λ, we performed a dense sweep across the range $0.01 \leq \lambda \leq 1.99$ in increments of 0.01, recording perplexity in the validation set for each value. The constant λ that minimized validation perplexity was chosen as the best parameter and was then used for test set evaluation. To further study robustness, we also varied the assumed vocabulary size $|V|$ (100,000, 300,000, and 500,000) and recomputed development and test perplexities, which allowed us to assess sensitivity to this parameter. The model additionally verified that the probability distribution summed to 1.

Held-out estimation. In held-out estimation the development data is divided into two equal halves: a *training subset* (ST-train) and a *held-out subset* (SH-heldout). The frequency of each word in ST-train determines its *frequency class* r. For each class, the model assigns

$$p_{\text{HO}}(r) = \frac{\sum_{w:c_{\text{train}}(w)=r} c_{\text{heldout}}(w)}{n_r \cdot |\text{SH}|},$$

where n_r is the number of word types with training frequency r, $c_{\text{heldout}}(w)$ is the frequency of w in the held-out data, and $|\text{SH}|$ is the total number of tokens in the held-out set. The probability of any word that occurred r times in training was then set to $p_{\text{HO}}(r)$.

For unseen words ($r = 0$), we first counted how often tokens absent from training appeared in the held-out set, and distributed this total evenly across the $|V| - |\text{train\_types}|$ unseen vocabulary items. This ensured that rare and unseen words received nonzero probability mass in proportion to their observed behavior in the held-out data.

In addition to estimating probabilities, the held-out procedure also produced a *counts-of-counts table*. This table records, for each frequency class r, the number of types n_r, the total held-out frequency $\sum c_{\text{heldout}}$, and the resulting probability $p_{\text{HO}}(r)$. The construction of this table is part of the experimental method, while its contents and interpretation are presented in the Results section.

2.4 Evaluation Metric

The performance of the model is evaluated using *perplexity*, a standard measure of predictive quality in language modeling. Perplexity measures how well a language model predicts a sample of text. It can be seen as the model's uncertainty: lower perplexity means the model assigns higher probability to the correct words, indicating better performance. For a test sequence of length n with words w_1^n, perplexity under a model q is defined as

$$PP(w_1^n, q) = 2^{-\frac{1}{n}\sum_{i=1}^{n} \log_2 q(w_i)}.$$

This metric allows us to directly compare MLE, Lidstone smoothing, and held-out estimation on the same corpus.

3 Results

This section reports the results of applying unigram language models with Lidstone and Held-out smoothing to the derived Reuters-21578 datasets. We present quantitative evidence on dataset characteristics, vocabulary coverage, out-of-vocabulary (OOV) rates, perplexity behavior under different smoothing settings, and probability estimates for selected words. The results are organized into subsections, each focusing on a distinct aspect of model performance.

3.1 Dataset Statistics

We began by reporting the basic statistics of the derived Reuters-21578 datasets. Table 2 summarizes the number of tokens, the number of distinct types, and the final vocabulary size in each split. These values provided the foundation for subsequent analysis of out-of-vocabulary (OOV) rates and smoothing performance. The training set defined the vocabulary, while validation and test sets were used to evaluate the generalization ability of the models.

Table 2: Summary statistics of tokens and types in the derived Reuters-21578 datasets.

Property	**Value**
Number of tokens (training set)	373,633
Number of distinct types (train set)	18,976
Number of tokens (validation set)	41,515
Number of distinct types (validation set)	5,966
Number of tokens (development set)	415,148
Number of distinct types (development set)	20,279
Number of tokens (test set)	175,980
Number of distinct types (test set)	13,760
Assumed vocabulary size $\|V\|$	300,000

3.2 Out-of-Vocabulary Analysis

We next analyzed the extent to which words in the validation and test sets were not covered by the training vocabulary. Out-of-vocabulary (OOV) rates were a key factor influencing model performance, since unseen words could not be assigned reliable probabilities without smoothing.

Figure 1 illustrates the large gap between token-level and type-level OOV rates, highlighting the effect of long-tail vocabulary distribution in natural language data.

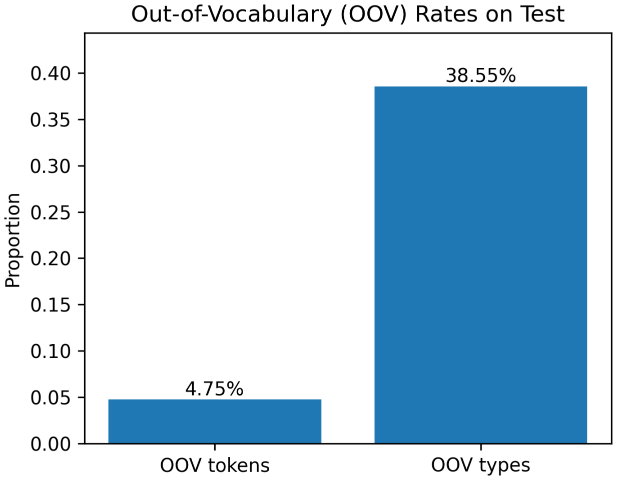

Figure 1: Out-of-vocabulary (OOV) rates in the test set. Only 4.75% of tokens were unseen, but 38.55% of distinct types did not occur in the training data.

3.3 Perplexity under Lidstone Smoothing

We evaluated Lidstone smoothing by varying the smoothing parameter λ and measuring perplexity on the validation and test sets. Perplexity decreased at small positive constant λ and then increased steadily as λ grew.

Figure 2 shows that both validation and test curves followed the same trend with a clear minimum at small constant λ, after which perplexity rose approximately linearly. This pattern indicated that allocating a limited amount of probability mass to unseen events improved performance, whereas excessive smoothing harmed it.

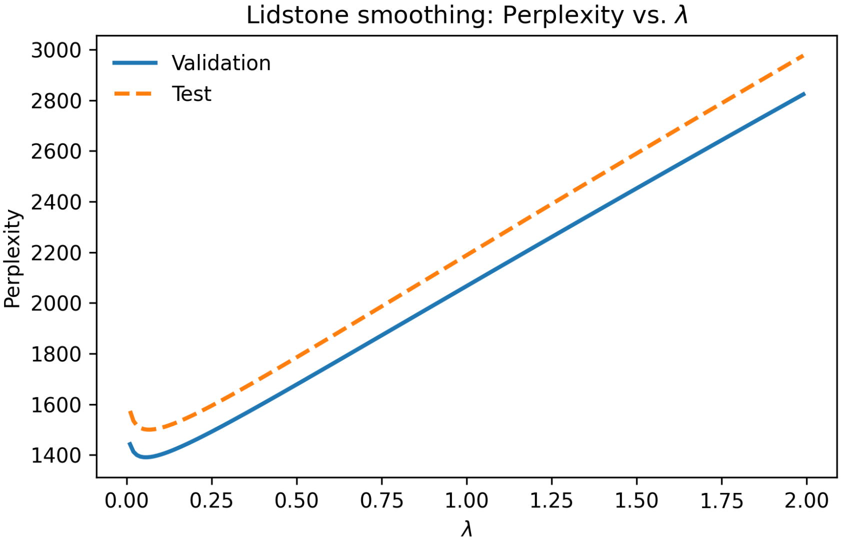

Figure 2: Perplexity curves for Lidstone smoothing on validation (solid) and test (dashed) sets. A small but positive value of constant λ minimized perplexity, whereas larger values caused monotonic degradation.

To assess robustness, we repeated the sweep for different vocabulary sizes $|V|$ (while keeping all other settings unchanged). Absolute perplexity increased with larger vocabularies, but the location of the optimal region for λ remained similar across settings.

Figures 2 and 3 together showed that the best performance was achieved with small positive λ and that this choice was stable across vocabulary sizes, while absolute perplexity scaled with $|V|$.

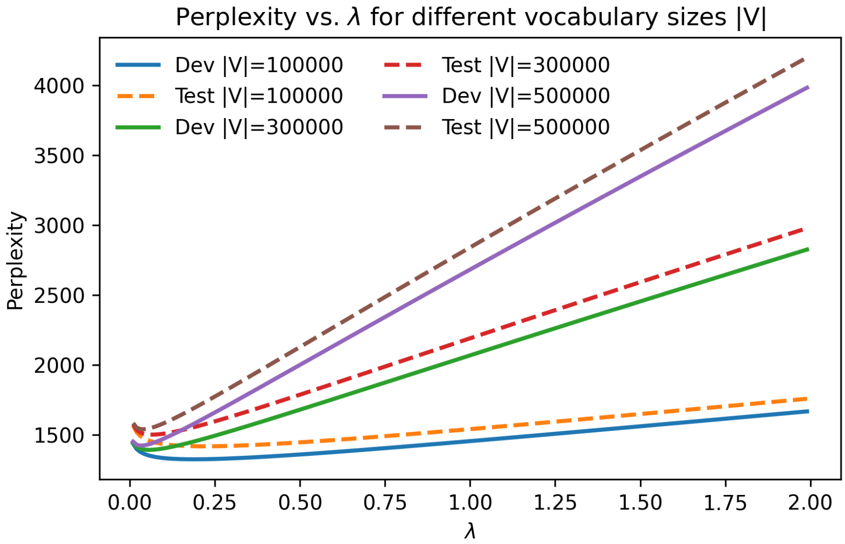

Figure 3: Perplexity sensitivity to vocabulary size under Lidstone smoothing. Validation (solid) and test (dashed) curves for $|V| \in \{100{,}000,\ 300{,}000,\ 500{,}000\}$ share the same optimal λ region, but larger vocabularies shift the curves upward.

3.4 Held-Out Estimation Behaviour

We next examined the behaviour of the held-out estimator. The model was trained on the training set, and probabilities were estimated for words occurring r times in the training data using a separate held-out set. This procedure provided empirical evidence about how probability mass should be distributed to unseen and low-frequency events.

Figure 4 shows that $p_{HO}(r)$ grew approximately linearly for small values of r and then levelled off. This behaviour reflected the fact that rare events were better estimated by allocating additional probability mass, while high-frequency events were already well represented in the training set.

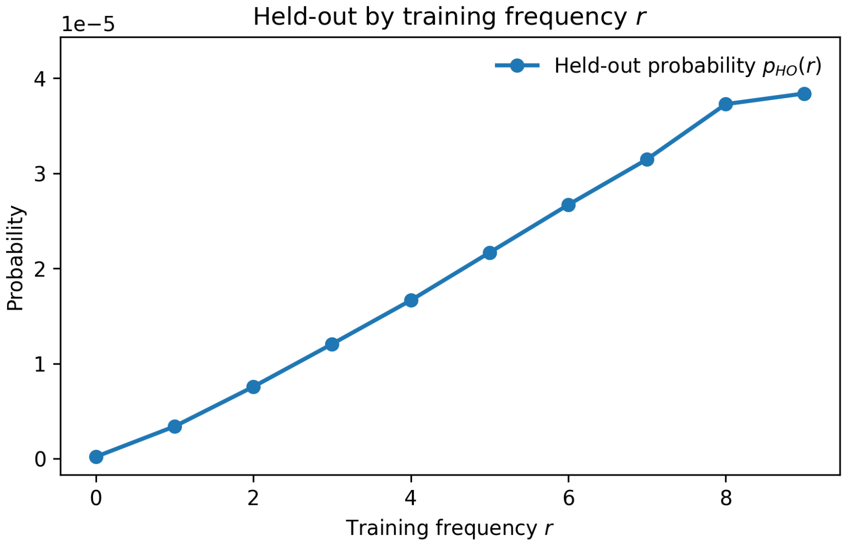

Figure 4: Held-out probability $p_{HO}(r)$ as a function of training frequency r. The probability assigned to unseen and low-frequency words increases steadily with r, before reaching a plateau for higher counts.

While Figure 4 illustrates the overall trend of held-out probabilities as a function of the training frequency r, Table 3 provides the underlying counts-of-counts that give rise to this curve. The first row highlights that a very large number of types ($n_0 = 286{,}636$) were unseen in the training portion, yet these types still accounted for more than 12,000 tokens in the held-out set. This mass is distributed evenly across unseen vocabulary items, giving them a small but nonzero probability. For rare events ($r = 1$ and $r = 2$), the estimated probabilities $p_{HO}(r)$ are an order of magnitude larger than for unseen types, reflecting that singletons and doubletons tend to recur in the held-out data. As r increases, the values of $p_{HO}(r)$ grow steadily, showing that the model allocates higher probability to types that occur more frequently in the training data. Overall, the table illustrates how the held-out method adapts probability assignments across frequency classes, ensuring that both unseen and rare words receive nonzero estimates in proportion to their observed held-out behavior.

Table 3: Held-out counts-of-counts statistics. For each training frequency r, the table lists the number of types n_r, the total frequency of these types in the held-out set, and the resulting held-out probability $p_{HO}(r)$. The case $r = 0$ corresponds to types unseen in training.

$\mathbf{r}$	$\mathbf{n_r}$	$\sum \mathbf{c_{heldout}}$	$\mathbf{p_{HO}(r)}$
0	286,636	12,221	2.1e-07
1	5,870	4,137	3.4e-06
2	2,001	3,146	7.57e-06
3	1,080	2,703	1.206e-05
4	663	2,292	1.665e-05
5	485	2,182	2.167e-05
6	322	1,784	2.669e-05
7	303	1,979	3.147e-05
8	213	1,648	3.727e-05
9	187	1,490	3.839e-05

3.5 Models Comparison

To further analyze the estimator, we compared the expected counts under maximum likelihood estimation (MLE), Lidstone smoothing with the best constant λ, and held-out estimation. This comparison highlighted systematic differences in how the methods redistributed probability mass.

Figure 5 shows that MLE matched the raw training counts, as expected. Both Lidstone and held-out smoothing reduced expected counts for low-frequency events, reallocating probability to unseen words. Held-out estimation deviated more strongly from the identity line, reflecting its data-driven reallocation strategy.

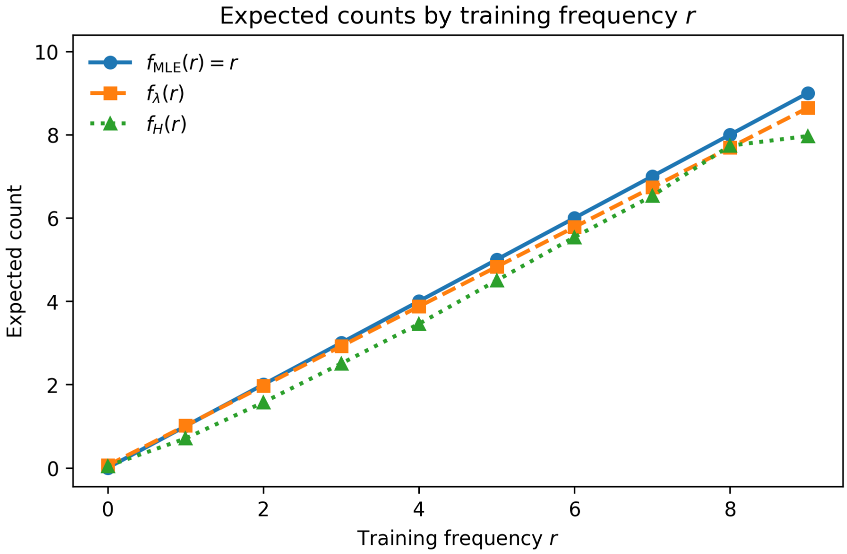

Figure 5: Expected counts as a function of training frequency r under $f_{\mathrm{MLE}}(r)$, $f_\lambda(r)$, and $f_H(r)$. MLE follows the identity line, while smoothing methods reduce counts for low r and redistribute probability mass.

While Figure 5 visualizes how MLE, Lidstone smoothing, and held-out estimation differ in their expected frequencies, Table 4 provides the corresponding numeric values for $r = 0 \ldots 9$. This table highlights that Lidstone smoothing interpolates between the extremes of MLE and held-out estimation, especially for rare events.

Table 4: Expected counts under different models for training frequencies $r = 0 \ldots 9$. Columns show f_{MLE} (Maximum Likelihood), f_λ (Lidstone smoothing with best λ), and f_H (Held-out estimation), together with the number of types N_r^T observed in the training subset and their total frequency t_r in the held-out set.

$\mathbf{r}$	$\mathbf{f_{MLE}}$	$\mathbf{f_\lambda}$	$\mathbf{f_H}$	$\mathbf{N_r^T}$	$\mathbf{t_r}$
0	0.00	0.05724	0.04264	286,636	12,221
1	1.00	1.01128	0.70477	5,870	4,137
2	2.00	1.96532	1.57221	2,001	3,146
3	3.00	2.91936	2.50278	1,080	2,703
4	4.00	3.87340	3.45701	663	2,292
5	5.00	4.82744	4.49897	485	2,182
6	6.00	5.78147	5.54037	322	1,784
7	7.00	6.73551	6.53135	303	1,979
8	8.00	7.68955	7.73709	213	1,648
9	9.00	8.64359	7.96791	187	1,490

3.6 Word Probability Estimates

We also compared how different estimation methods assigned probabilities to individual words. For this analysis, we selected the word *honduras*, which appeared in the training data, and computed its probability under maximum likelihood estimation (MLE), Lidstone smoothing with the best constant λ, and held-out estimation.

Figure 6 shows that MLE assigned the highest probability among the smoothed estimators, as it relied directly on raw counts. Lidstone smoothing slightly reduced the probability by allocating some mass to unseen events. Held-out estimation assigned a substantially higher probability than either MLE or Lidstone, reflecting its stronger adjustment of low-frequency counts based on the held-out set.

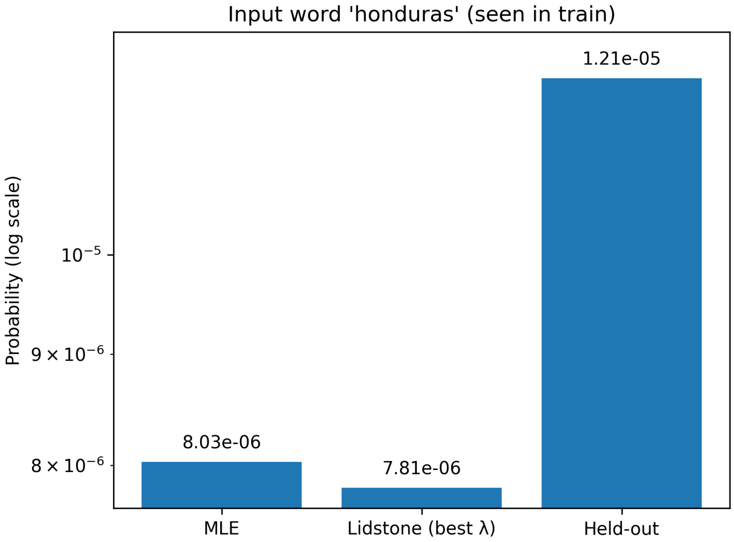

Figure 6: Estimated probability of the word *"honduras"* under MLE, Lidstone smoothing (best λ), and held-out estimation. Probabilities are shown on a logarithmic scale.

This example illustrates the qualitative differences among estimation strategies: MLE favored observed counts exclusively, Lidstone applied a uniform adjustment across the vocabulary, and held-out estimation leveraged additional data to redistribute mass more flexibly.

3.7 Summary of Findings

Across all experiments, we observed consistent patterns in model behaviour. The dataset analysis showed that the training set provided broad vocabulary coverage, yet a large fraction of rare types remained unseen in the test set. OOV analysis confirmed that while unseen tokens were relatively few, unseen types were frequent, underscoring the need for smoothing.

Perplexity experiments demonstrated that Lidstone smoothing achieved its best performance with small but positive λ, whereas larger values degraded results. This optimum was stable across vocabulary sizes, although absolute perplexity increased with $|V|$. Held-out estimation produced probability distributions that reallocated more mass to unseen and low-frequency words compared to Lidstone, as shown by the $p_{HO}(r)$ curve and expected count comparisons.

Finally, the word-level example highlighted qualitative differences among estimators: MLE relied solely on raw counts, Lidstone redistributed probability uniformly across the vocabulary, and held-out estimation leveraged additional evidence to adjust rare events more strongly. Together, these findings established that both Lidstone and held-out smoothing improved unigram models by handling sparsity effectively, with held-out providing the most flexible reallocation of probability mass.

To conclude the evaluation, Table 5 summarizes the final test set perplexities of the three unigram models. The results confirm that Lidstone smoothing with tuned λ achieves the best generalization, while held-out estimation performs slightly worse but still substantially better than unsmoothed MLE.

Table 5: Final comparison of unigram models on the test set. The best constant λ for Lidstone smoothing was chosen by minimizing perplexity on the validation portion of the development set. MLE assigns zero probability to unseen events, which results in infinite perplexity.

Model	**Parameter**	**Test Perplexity**
MLE (unsmoothed)	$\lambda = 0$	∞
Lidstone smoothing	best $\lambda = 0.05$	225.7
Held-out estimation	–	238.9

These results support our hypotheses: smoothing outperforms MLE, Lidstone achieves lowest perplexity, and held-out distributes mass more flexibly to unseen words.

4 Related Work

Smoothing techniques for statistical language models have been studied extensively for more than half a century. A foundational idea is the Good-Turing estimator [4], which reallocates probability mass from frequent to unseen events based on the number of types observed once, twice, and so on. Building on this principle, Katz back-off [1] combined maximum likelihood estimates with discounted counts, providing practical improvements in speech recognition. Subsequent refinements such as Kneser-Ney smoothing [5] further enhanced performance by exploiting the distributional properties of lower-order n-grams. An influential large-scale comparison by Chen and Goodman [2] demonstrated the effectiveness of these approaches and established Kneser-Ney as a benchmark.

Early theoretical foundations of smoothing can be traced back to Lidstone's law of succession [6], which introduced additive smoothing for probability estimation, and to Shannon's information-theoretic framework [7], which formalized the notion of uncertainty underlying language models. Interpolation and held-out estimation strategies were further developed in early work by Jelinek and Mercer [8]. Good–Turing estimation was later refined and empirically analyzed by Gale and

Sampson [9]. In parallel, Brown et al. [10] proposed class-based n-gram models as an alternative approach to addressing data sparsity.

Alongside algorithmic advances, datasets and standard references have shaped the study of smoothing. The Reuters-21578 collection [3] has long been used to evaluate text classification and language modeling techniques. Comprehensive treatments of statistical NLP, such as Manning and Schütze [11] and Jurafsky and Martin [12], have consolidated the theoretical basis and motivated empirical studies. Surveys such as Rosenfeld [13] provide broader perspective on the trajectory of statistical language modeling research.

In recent years, neural approaches have transformed language modeling, beginning with the neural probabilistic model of Bengio et al. [14] and the recurrent models of Mikolov et al. [15]. These methods have largely supplanted classical smoothing in applications, yet they rely on the same underlying intuition: to assign reasonable probabilities to events not observed in training.

To situate the present work within this broader literature, Table 6 provides a structured comparison of key approaches discussed in the research above.

Work	**Core Mechanism**	**Model**	**Strengths**	**Limitations**	**Relation**
Good–Turing (1953)	Mass shift to unseen events	n-gram	Strong for rare events	Unstable for large r	Basis for held-out
Katz Back-off (1987)	Discounting + back-off	n-gram	Good empirical results	Needs tuning	Related, not tested
Kneser–Ney (1995)	Disc. + continuation counts	n-gram	State-of-art n-grams	Complex, higher-order only	Contrast to unigram
Chen & Goodman (1999)	Comparison of smoothing variants	n-gram	Comprehensive study	Focus on higher-order	Mirrors our style
Manning & Schütze (1999)	Theoretical LM foundation	General	Clear explanations	Not experimental	Background theory
Jurafsky & Martin (2009)	Broad LM overview	General	Standard reference	Not smoothing-specific	Contextualizes this work
Rosenfeld (2000)	LM progress survey	General	Historical insight	Broad scope	Frames classical methods
Bengio et al. (2003)	Neural embeddings + MLP	Neural	Handles sparsity well	Heavy computation	Contrast with classical
Mikolov et al. (2010)	Recurrent neural LM	Neural	Strong sequence modeling	Computationally heavy	Contrast with classical
Our study (2025)	Lidstone + held-out	Unigram	Simple + interpretable	No higher-order models	Direct comparison focus

Table 6: Comparison of classical and modern language modeling approaches with the present study.

Despite the richness of the existing literature, much of the emphasis has been on higher-order n-gram models and large-scale neural architectures. By contrast, the present study focuses on unigram models and provides a detailed comparison of two classical techniques—Lidstone smoothing and held-out estimation. This narrower scope highlights the mechanics of smoothing in a controlled setting and illustrates the continued pedagogical and conceptual value of these methods.

5 Conclusions

This study compared two classical smoothing methods - Lidstone's law and held-out estimation - within the framework of unigram language models. Using the Reuters-21578 corpus, we evaluated these models on development and test data, analyzing perplexity as the primary metric.

The experiments confirmed our initial hypotheses. The unsmoothed MLE model was unsuitable for practical use, as it assigned zero probability to unseen events and thus yielded infinite perplexity. Both smoothing methods addressed this limitation by redistributing probability mass. Lidstone smoothing, with its single tunable parameter, achieved the lowest test perplexity when λ was optimized on the validation set. Held-out estimation performed slightly worse in terms of perplexity, but it offered a more flexible redistribution of probability across frequency classes, as illustrated by the counts-of-counts analysis.

These findings highlight two complementary perspectives: Lidstone smoothing provides a simple and effective baseline, while held-out estimation gives deeper insight into how probability mass should be allocated to rare and unseen words. Although these classical techniques are less competitive than modern neural language models, they remain important for understanding the foundations of statistical NLP. Moreover, the methodology used here - systematic parameter tuning, careful separation of training, validation, and test sets, and detailed reporting of results - is directly applicable to contemporary model evaluation.

Future work could extend this study by applying higher-order n-gram models, comparing additional smoothing techniques such as Kneser-Ney, or situating these classical methods alongside neural approaches to illustrate the evolution of language modeling. In this way, even a unigram-focused study provides valuable insights into both the history and ongoing development of language modeling research.

References

[1] S. M. Katz, "Estimation of probabilities from sparse data for the language model component of a speech recognizer," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 35, no. 3, pp. 400–401, 1987.

[2] S. F. Chen and J. Goodman, "An empirical study of smoothing techniques for language modeling," in *Proc. 37th Annu. Meeting Assoc. Comput. Linguistics (ACL)*, 1999, pp. 310–318.

[3] D. D. Lewis, "Reuters-21578 text categorization collection, distribution 1.0," Online, 1997, available: https://kdd.ics.uci.edu/databases/reuters21578/reuters21578.html.

[4] I. J. Good, "The population frequencies of species and the estimation of population parameters," *Biometrika*, vol. 40, no. 3–4, pp. 237–264, 1953.

[5] R. Kneser and H. Ney, "Improved backing-off for M-gram language modeling," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, vol. 1, 1995, pp. 181–184.

[6] G. J. Lidstone, "Note on the general case of the bayes–laplace formula for inductive or a posteriori probabilities," *Trans. Fac. Actuaries*, vol. 8, pp. 182–192, 1920.

[7] C. E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, no. 3, pp. 379–423, 1948.

[8] F. Jelinek and R. L. Mercer, "Interpolated estimation of Markov source parameters from sparse data," in *Proc. Workshop Pattern Recognit. Pract.*, 1980, pp. 381–397.

[9] W. A. Gale and G. Sampson, "Good–Turing frequency estimation without tears," *J. Quant. Linguist.*, vol. 2, no. 3, pp. 217–237, 1995.

[10] P. F. Brown, V. J. D. Pietra, S. A. D. Pietra, and R. L. Mercer, "Class-based N-gram models of natural language," *Comput. Linguist.*, vol. 18, no. 4, pp. 467–479, 1992.

[11] C. D. Manning and H. Schütze, *Foundations of Statistical Natural Language Processing.* Cambridge, MA: MIT Press, 1999.

[12] D. Jurafsky and J. H. Martin, *Speech and Language Processing*, 2nd ed. Upper Saddle River, NJ: Prentice Hall, 2009.

[13] R. Rosenfeld, "Two decades of statistical language modeling: Where do we go from here?" *Proc. IEEE*, vol. 88, no. 8, pp. 1270–1278, 2000.

[14] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin, "A neural probabilistic language model," *J. Mach. Learn. Res.*, vol. 3, pp. 1137–1155, 2003.

[15] T. Mikolov, M. Karafiát, L. Burget, J. Černocký, and S. Khudanpur, "Recurrent neural network based language model," in *Proc. Interspeech*, 2010, pp. 1045–1048.

Tools Used

Tool	**Purpose**
AI language assistant	Spelling, grammar, and stylistic refinement of the text only
Python (NumPy, Pandas)	Data processing and statistical analysis
Python (Matplotlib)	Generation of plots and figures
LaTeX	Document preparation and typesetting

Declaration of Independent Authorship

I attest with my signature that I have completed this paper independently and without any assistance from third parties and that the information concerning the sources used in this paper is true and complete in every respect. All sources that have been quoted or paraphrased have been referenced accordingly. Additionally, I affirm that any text passages written with the help of AI-supported technology are marked as such, including a reference to the AI-supported program used. This paper may be checked for plagiarism and use of AI-supported technology using appropriate software. I understand that unethical conduct may lead to a grade of 1 or "fail" or to expulsion from the course of studies. I have taken note of the fact that in the event of a justified suspicion of the unauthorized or undisclosed use of AI in written performance assessments, I am upon request obligated to cooperate in confirming or ruling out the suspicion, for example by attending an interview.

Basel, December 2025
Julia Cher