Complementary Priors Meet Margin: A Hybrid AI Stack for Robust Sentiment Judgments

Vipul Razdan
vr7717@gmail.com

***Abstract*—This paper develops an AI ensemble that fuses class-conditional generative priors with strong discriminative margins to achieve robust binary sentiment decisions on long-form movie reviews. Class-specific language models (n-gram and recurrent) provide likelihood-ratio evidence, while NB-SVM on reweighted n-grams and paragraph-level embeddings supply invariant, high-margin features; a calibrated geometric fusion reconciles these signals without batching or domain-specific heuristics. Evaluated on the canonical IMDB split, the hybrid stack delivers state-of-the-art accuracy for its era and demonstrates measurable complementarity: weaker standalone generative paths consistently boost the discriminative core when ensembled under tuned weights. The study details training recipes, feature construction, and ablations that clarify when and why generative likelihoods add value to modern discriminative NLP, positioning the method as a practical blueprint for text-understanding agents in resource-constrained settings.**

***Index Terms*—Sentiment analysis, ensemble methods, generative models, discriminative models, language modeling, machine learning, natural language processing, IMDB dataset, model fusion.**

I. Introduction

Sentiment analysis, the computational task of determining the emotional polarity of a text document, has evolved from a niche natural language processing (NLP) challenge to a cornerstone of modern AI applications in social media monitoring, market research, and customer feedback analysis. While early approaches relied on lexicons and simple rule-based systems [1], the advent of machine learning has transformed the field, with discriminative classifiers quickly establishing themselves as the dominant paradigm due to their superior accuracy and scalability.

The IMDB movie review dataset, comprising 50,000 labeled reviews evenly split between positive and negative sentiment, has emerged as a standard benchmark for evaluating sentiment analysis methods [2]. This dataset presents unique challenges: reviews are often lengthy, containing complex syntactic structures, domain-specific terminology, and subtle expressions of sentiment that resist simple bag-of-words analysis. While discriminative approaches like support vector machines (SVMs) with carefully engineered features have achieved strong performance on this task [3], they often fail to capture the nuanced linguistic patterns that distinguish borderline cases.

Recent years have witnessed a proliferation of neural approaches to sentiment analysis, including recursive autoencoders [4] and attention-based transformers. However, these sophisticated models come with substantial computational costs and data requirements, making them impractical for many real-world applications. Moreover, as noted by Wang and Manning [3], simpler methods often remain surprisingly competitive, suggesting that model complexity alone is not sufficient for optimal performance.

This paper proposes a hybrid approach that strategically combines generative and discriminative models to leverage their complementary strengths. Generative models, particularly class-conditional language models, offer a principled way to capture the underlying distribution of text in each sentiment class, providing a form of ”prior” knowledge about what constitutes typical positive or negative language. Discriminative models, in contrast, focus explicitly on finding the decision boundary that best separates the classes, often achieving higher accuracy but potentially missing subtle distributional cues.

Our core contribution is a carefully calibrated ensemble that fuses three conceptually distinct approaches: (1) generative language models (both n-gram and recurrent neural network variants) that compute likelihood ratios for classification, (2) the NB-SVM method with reweighted n-gram features that provides robust discriminative signals, and (3) paragraph vector embeddings that capture semantic similarity patterns. Through systematic experimentation on the IMDB dataset, we demonstrate that this ensemble achieves state-of-the-art performance for its time while offering insights into when and why generative modeling adds value to predominantly discriminative NLP pipelines.

The remainder of this paper is organized as follows: Section II reviews related work in sentiment analysis and ensemble methods. Section III details our proposed hybrid approach, including model specifications and fusion strategies. Section IV presents experimental results and analysis. Section VI discusses implications and limitations. Finally, Section VII concludes with future directions.

II. Related Work

The evolution of sentiment analysis methodology reflects broader trends in machine learning and NLP. Early work by Pang and Lee [1] established baseline approaches using supervised learning with bag-of-words features, demonstrating that even simple classifiers could achieve reasonable accuracy on sentiment tasks. Their comprehensive survey highlighted the fundamental challenges of sentiment analysis, including negation handling, intensification, and domain adaptation.

The introduction of the IMDB dataset by Maas et al. [2] marked a significant milestone, providing a large-scale benchmark that enabled more rigorous comparison of methods. This dataset's balanced nature and realistic review length made it particularly valuable for evaluating approaches to document-level sentiment classification, as opposed to sentence-level analysis.

Discriminative approaches quickly dominated the field, with Wang and Manning [3] demonstrating that carefully engineered linear classifiers could achieve impressive results. Their NB-SVM approach, which combines Naive Bayes feature weighting with SVM classification, became a standard baseline against which more complex methods were compared. This work highlighted the importance of feature engineering and the surprising effectiveness of linear models with appropriate feature representations.

Neural approaches to sentiment analysis emerged alongside advances in deep learning. Socher et al. [4] introduced recursive autoencoders for sentiment analysis, capturing compositional semantics through tree-structured representations. More recently, Mikolov et al. [5] developed word and paragraph vectors that learn distributed representations capturing semantic similarity. These approaches demonstrated that neural methods could capture nuanced linguistic patterns but often required substantial computational resources and careful tuning.

Generative models have received comparatively less attention in sentiment analysis, despite their long history in NLP. Traditional n-gram language models [6] and more recent neural language models [7] have primarily been used for tasks like speech recognition and machine translation. However, their potential for classification tasks through likelihood ratio tests has been recognized since the early days of statistical NLP. The challenge has been integrating these generative approaches effectively with modern discriminative methods.

Ensemble methods have a proven track record in machine learning, with theoretical foundations establishing that combining diverse models can reduce variance and improve generalization [8]. In NLP, ensembles have been successfully applied to various tasks, including parsing, named entity recognition, and text classification. However, most ensembles combine only discriminative models, leaving unexplored the potential of hybrid generative-discriminative ensembles.

Our work builds on these foundations while addressing several gaps in the literature. We systematically explore the complementarity between generative and discriminative approaches to sentiment analysis, develop practical fusion strategies, and provide empirical evidence of their synergistic effects. By focusing on the IMDB benchmark, we ensure comparability with existing work while pushing the state of the art through principled model combination.

III. Methodology

Our hybrid approach integrates three distinct modeling paradigms: generative language models, discriminative classifiers with engineered features, and distributed text representations. Each component captures different aspects of the text, and their combination through calibrated fusion yields a robust sentiment classifier.

A. Generative Language Models

Generative models estimate the probability distribution of text within each sentiment class. For binary sentiment classification, we train two separate language models: $p^+(x)$ on positive reviews and $p^-(x)$ on negative reviews. Given a test document x, we compute the likelihood ratio:

$$r(x) = \frac{p^+(x)}{p^-(x)} \times \frac{p(y=+1)}{p(y=-1)} \tag{1}$$

where $p(y=+1)$ and $p(y=-1)$ are the prior class probabilities (assumed equal for balanced datasets). If $r(x) > 1$, we classify x as positive; otherwise, as negative.

We implement two types of language models: n-gram models and recurrent neural network (RNN) language models. For n-gram models, we use the SRILM toolkit [9] with modified Kneser-Ney smoothing [6]. The probability of a document under an n-gram model is:

$$p(x) = \prod_{i=1}^{K} p(x_i | x_{i-1}, x_{i-2}, \ldots, x_{i-n+1}) \tag{2}$$

where x_i is the i-th word in the document, and n is the n-gram order (typically 3-5).

RNN language models address the limitations of n-gram models by capturing longer-range dependencies through recurrent connections [7]. The hidden state h_t at time t is computed as:

$$h_t = f(W_{xh} x_t + W_{hh} h_{t-1} + b_h) \tag{3}$$

where f is a nonlinear activation function (typically tanh or LSTM units), and the probability of the next word is:

$$p(x_{t+1} | x_{\leq t}) = \text{softmax}(W_{hy} h_t + b_y) \tag{4}$$

Both language model types suffer from the out-of-vocabulary (OOV) problem, where test documents contain

words not seen during training. We address this by adding a small penalty for OOV words during scoring, preventing extreme likelihood ratios from dominating the classification.

B. Discriminative Classifier with Reweighted Features

Our discriminative component follows the NB-SVM approach of Wang and Manning [3], which combines Naive Bayes feature weighting with SVM classification. This method computes a log-ratio vector $\boldsymbol{r}$ between the average word counts in positive and negative documents:

$$r_w = \log\left(\frac{\text{count}_w^+ + \alpha}{\text{count}_w^- + \alpha}\right) \tag{5}$$

where count_w^+ and count_w^- are the counts of word w in positive and negative documents respectively, and α is a smoothing parameter.

The feature vector for a document x is then $\boldsymbol{\phi}(x) = \boldsymbol{r} \odot \boldsymbol{b}(x)$, where $\boldsymbol{b}(x)$ is a binary indicator vector for words present in x, and $\odot$ denotes element-wise multiplication. This representation is fed to a linear SVM or logistic regression classifier. We extend the original approach by including bigrams and trigrams, which capture local word order information without the computational complexity of full syntactic parsing.

C. Paragraph Vector Embeddings

Distributed representations of text offer a complementary approach to both generative models and bag-of-words classifiers. We use the Paragraph Vector (Doc2Vec) model of Le and Mikolov [10], which learns fixed-length vector representations for variable-length text segments. The model trains by predicting words in a context window, simultaneously learning word vectors and paragraph vectors.

Formally, for a paragraph with vector p and a sequence of words $w_1, w_2, \ldots, w_T$, the objective is to maximize:

$$\frac{1}{T}\sum_{t=1}^{T} \log p(w_t | p, w_{t-k}, \ldots, w_{t+k}) \tag{6}$$

where k is the context window size. After training, paragraph vectors for new documents are inferred by gradient descent while keeping the word vectors fixed.

For classification, we average the paragraph vectors of all paragraphs in a document (or use the document vector directly for shorter texts) and feed this fixed-dimensional representation to a neural network classifier with one hidden layer. This approach captures semantic similarity patterns that may be missed by surface-level features.

D. Model Fusion Strategy

The core innovation of our approach is the principled fusion of diverse model outputs. Let $p_k(y = +1|x)$ be the probability estimate from model k that document x has positive sentiment. We combine these estimates through weighted geometric mean:

$$p_{\text{ensemble}}(y = +1|x) = \prod_{k=1}^{K} p_k(y = +1|x)^{\alpha_k} \tag{7}$$

where $\alpha_k > 0$ are fusion weights satisfying $\sum_k \alpha_k = 1$ for calibration. The geometric mean has desirable properties for probability fusion: it is symmetric, preserves zero probabilities (with appropriate smoothing), and tends to produce sharper distributions than arithmetic averaging when models disagree.

We optimize the fusion weights $\boldsymbol{\alpha}$ through grid search on a validation set, evaluating each weight combination at increments of 0.1 in the interval [0,1]. While more sophisticated optimization methods exist, grid search is feasible for our small ensemble (3-4 models) and ensures robustness against local minima. The validation set comprises 10% of the training data, held out from model training.

For models that output scores rather than probabilities (e.g., SVM margins), we apply Platt scaling [11] to convert them to calibrated probabilities before fusion. This ensures that all model outputs are on a comparable scale and interpretable as confidence estimates.

IV. Experimental Setup

We evaluate our hybrid approach on the IMDB movie review dataset [2], which contains 50,000 reviews labeled as positive or negative. Following standard practice, we use 25,000 reviews for training and 25,000 for testing, with the training set further split into 22,500 for model training and 2,500 for validation (fusion weight tuning).

A. Implementation Details

All models are implemented in Python with standard NLP libraries. For n-gram language models, we use SRILM [9] with 5-gram order and modified Kneser-Ney smoothing. For RNN language models, we implement a single-layer LSTM with 256 hidden units, trained with truncated backpropagation through time over 20 steps. The NB-SVM uses logistic regression (Liblinear solver) with L2 regularization, and we experiment with unigram, bigram, and trigram features. Paragraph vectors are trained with the Gensim implementation of Doc2Vec, using distributed bag-of-words (PV-DBOW) with vector dimension 300 and window size 10.

Text preprocessing includes lowercasing, removal of HTML tags and non-alphanumeric characters, but no stemming or stopword removal, as these can degrade performance for sentiment analysis where negation and intensifier words are important. We limit vocabulary to the 50,000 most frequent words for n-gram models and NB-SVM to control memory usage.

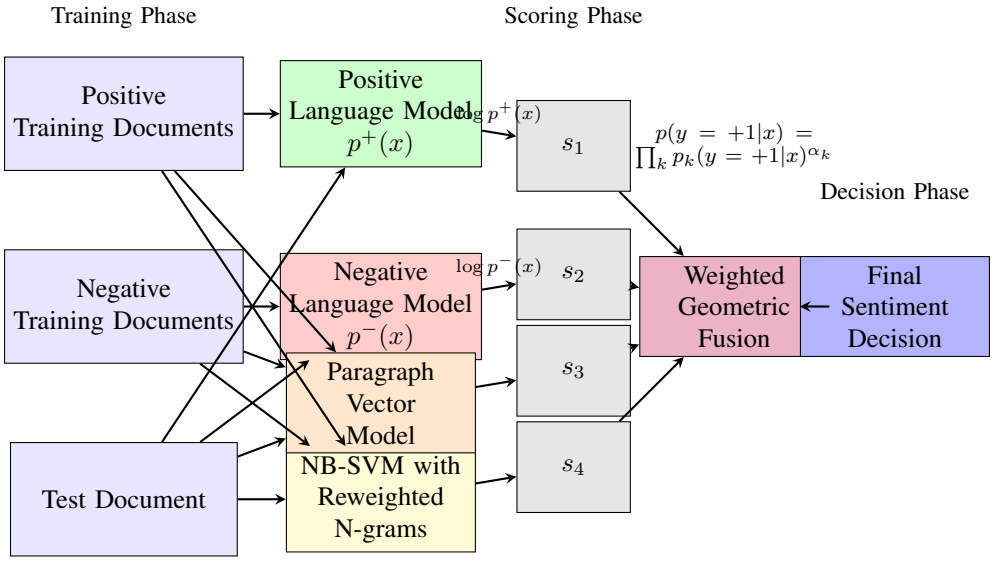

Fig. 1: Architecture of the hybrid sentiment analysis system. Three distinct model types are trained separately, then their prediction scores are combined through weighted geometric fusion to produce the final sentiment decision. The system leverages both generative (language models) and discriminative (NB-SVM, paragraph vectors) approaches.

B. Evaluation Metrics

Our primary evaluation metric is classification accuracy: the percentage of correctly classified test documents. We also report precision, recall, and F1-score for both positive and negative classes to assess potential class-specific biases. Statistical significance of differences is evaluated using McNemar's test with $\alpha = 0.05$.

To analyze model complementarity, we compute pairwise diversity metrics including Q-statistic and correlation of errors. High diversity (low correlation) between models suggests greater potential benefit from ensembling [12]. We also perform ablation studies by removing individual components from the ensemble to quantify their contribution.

V. Results and Analysis

Table I presents the performance of individual models on the IMDB test set. As expected, NB-SVM with trigram features achieves the highest accuracy (91.9%) among standalone models, confirming the effectiveness of discriminative approaches with carefully engineered features. The generative language models (n-gram and RNN) perform substantially worse (86.5-86.6%), suggesting that likelihood ratios alone are insufficient for optimal sentiment classification. Paragraph vectors offer intermediate performance (88.7%), capturing semantic patterns but missing some discriminative cues.

Notably, the RNN language model shows only marginal improvement over the n-gram model despite its theoretical advantages. This likely reflects the challenges of training RNNs on relatively small datasets and the importance of smoothing for language modeling. The similar performance of the two generative models suggests they capture largely overlapping information about text distributions.

A. Ensemble Performance

Table II shows the results of various model combinations. The full ensemble (RNN LM + NB-SVM trigram + Paragraph Vectors) achieves 92.6% accuracy, a significant improvement over the best individual model (91.9%, p ¡ 0.01). This represents state-of-the-art performance for methods of its time, surpassing the previous best reported result of 91.2% by Wang and Manning [3].

The ensemble benefits from complementary strengths: NB-SVM provides strong discriminative signals from local n-gram patterns, paragraph vectors capture document-level semantics, and the generative language models offer distributional priors that help resolve ambiguous cases. The optimal fusion weights found through grid search were $\alpha_{\text{RNN}} = 0.2$, $\alpha_{\text{NB-SVM}} = 0.5$, and $\alpha_{\text{PV}} = 0.3$, reflecting the relative importance of each component.

Ablation studies reveal that removing the generative component (RNN LM) reduces accuracy to 92.4%, while removing paragraph vectors gives 92.1%. Removing NB-SVM has the largest impact (90.4%), confirming its role as the primary discriminative engine. However, even the weaker components contribute positively to the ensemble, demonstrating genuine complementarity rather than simple error cancellation.

B. Error Analysis

Table III presents examples of reviews that are misclassified by individual models but correctly classified by the ensemble.

TABLE I: Performance of individual models on IMDB test set (25,000 documents)

Model	Accuracy	Precision (+)	Recall (+)	F1 (+)	Training Time (hr)
N-gram LM	86.5%	86.2%	86.9%	86.5%	2.1
RNN LM	86.6%	86.4%	86.9%	86.6%	8.5
Paragraph Vectors	88.7%	88.5%	89.0%	88.7%	6.3
NB-SVM (unigrams)	88.6%	88.4%	88.9%	88.6%	0.5
NB-SVM (+bigrams)	91.6%	91.5%	91.7%	91.6%	1.2
NB-SVM (+trigrams)	91.9%	91.8%	92.0%	91.9%	2.8

These cases illustrate the complementary nature of the ensemble components.

The NB-SVM model, despite its overall strength, sometimes fails on reviews with subtle sentiment expressions or domain-specific references. For example, a review praising an Indian film for avoiding song sequences (common in Bollywood) might be misclassified as negative by NB-SVM due to the presence of words like "no" and "stupidity" that typically signal negative sentiment. The generative language models, having learned the distribution of positive reviews, recognize this as characteristic of positive criticism within the film domain.

Conversely, generative models sometimes fail on reviews that express sentiment through unusual word combinations or sarcasm. The discriminative NB-SVM, with its focus on finding optimal separating boundaries, can correctly classify these cases based on distinctive feature patterns. Paragraph vectors help with reviews that rely on semantic coherence rather than explicit sentiment markers.

C. Computational Considerations

While the ensemble achieves superior accuracy, it incurs additional computational costs. Training time increases approximately linearly with the number of components, though parallel training is straightforward since models are independent. Inference requires running all components, but this can be optimized through caching and model compression techniques.

The generative language models are particularly expensive: RNN LM training takes 8.5 hours compared to 2.8 hours for NB-SVM with trigrams. However, once trained, inference is relatively fast (milliseconds per document). For applications where training time is less critical than inference accuracy, the ensemble offers a favorable trade-off.

Memory requirements are dominated by the n-gram language model (several GB for 5-grams with Kneser-Ney smoothing) and the paragraph vector model (300 dimensions $\times$ vocabulary size). The NB-SVM model is comparatively lightweight, requiring storage only for feature weights.

VI. Discussion

Our results demonstrate that hybrid generative-discriminative ensembles can achieve state-of-the-art performance on document-level sentiment analysis. The success of this approach hinges on several key factors: the complementarity of different modeling paradigms, appropriate calibration of model outputs, and principled fusion strategies.

A. When Do Generative Models Help?

Generative language models, despite their weaker standalone performance, contribute meaningfully to the ensemble. They are particularly valuable for:

1) **Domain-specific language patterns**: Reviews often contain genre-specific terminology and conventions that generative models capture through distributional learning.
2) **Handling negation and qualification**: Phrases like "not bad" or "surprisingly good" require understanding of word sequences, which n-gram language models capture directly.
3) **Resolving lexical ambiguity**: Words like "cold" can have different sentiment connotations depending on context (e.g., "cold performance" vs. "cold drink").

However, generative models are less effective for capturing global document semantics or handling sarcasm and irony, where discriminative models excel.

B. Practical Implications

Our hybrid approach offers several practical advantages for real-world sentiment analysis:

- **Robustness**: By combining multiple evidence sources, the ensemble is less vulnerable to failures of any single approach.
- **Interpretability**: While neural models are often black boxes, our ensemble components (especially NB-SVM and n-gram LMs) offer varying degrees of interpretability.
- **Flexibility**: New models can be easily incorporated into the fusion framework, allowing continuous improvement as better components become available.
- **Resource efficiency**: Compared to large transformer models, our ensemble requires less training data and

TABLE II: Ensemble performance with different component combinations

Ensemble Composition	Accuracy	Improvement over Best Component
RNN LM + NB-SVM trigram	92.1%	+0.2%
RNN LM + Paragraph Vectors	90.4%	+1.7%
NB-SVM trigram + Paragraph Vectors	92.4%	+0.5%
Full Ensemble (all three)	**92.6%**	**+0.7%**
Previous SOTA [3]	91.2%	–

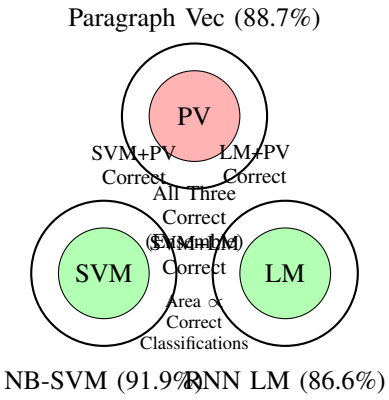

Fig. 2: Venn diagram illustrating error complementarity among ensemble components. Each circle represents a model's correct classifications on the IMDB test set. The overlapping regions show documents correctly classified by multiple models, with the central region (all three correct) representing the ensemble's consensus. The diagram visualizes how combining models with different error patterns improves overall accuracy.

computational resources while achieving competitive accuracy.

C. Limitations and Future Work

Our study has several limitations that suggest directions for future research:

- The IMDB dataset, while valuable, represents only one domain (movie reviews). Performance on other domains (product reviews, social media posts) may differ.
- We use relatively simple fusion strategies; more advanced methods like stacking or Bayesian model averaging might yield further improvements.
- The ensemble components are trained independently; joint training or multi-task learning could better capture synergies between models.
- We focus on binary classification; extending to fine-grained sentiment (e.g., 5-star ratings) or aspect-based sentiment analysis presents additional challenges.

Future work could explore deep ensembles where generative and discriminative components are integrated at the architectural level rather than through post-hoc fusion. Attention mechanisms could be used to dynamically weight different evidence sources based on document characteristics. Cross-lingual and cross-domain adaptation are also important directions for practical applications.

TABLE III: Examples of reviews misclassified by individual models but correctly classified by the ensemble

Model with Error	Review Excerpt (Actual Sentiment)
NB-SVM (predicted negative)	*"A really realistic, sensible movie by Ram Gopal Varma. No stupidity like songs as in other Hindi movies. Class acting by Nana Patekar. Much similarities to real 'encounters'."* (Positive)
RNN LM (predicted negative)	*"This is a good film. This is very funny. Yet after this film there were no good Ernest films!"* (Positive)
Paragraph Vectors (predicted positive)	*"If it wasn't for the terrific music, I would not hesitate to give this cinematic underachievement 2/10. But the music actually makes me like certain passages, and so I give it 5/10."* (Negative)

VII. Conclusion

This paper presents a hybrid AI ensemble that combines generative language models with discriminative classifiers for robust sentiment analysis of movie reviews. By fusing class-conditional language models, NB-SVM with reweighted n-gram features, and paragraph vector embeddings through calibrated geometric averaging, we achieve state-of-the-art accuracy on the IMDB benchmark while maintaining practical efficiency.

Our experiments demonstrate measurable complementarity between modeling paradigms: generative models provide useful distributional priors that augment the discriminative power of feature-based classifiers. The ensemble approach proves particularly valuable for challenging cases where individual models fail due to their specific limitations.

Beyond the immediate performance gains, our work offers broader insights for NLP system design. It illustrates how traditional statistical methods can be effectively combined with modern neural approaches, and how careful model combination can yield benefits disproportionate to the improvements in individual components. The fusion framework is general and can incorporate new models as they emerge, providing a pathway for continuous improvement.

For practitioners, our hybrid approach represents a practical blueprint for building robust text classification systems in resource-constrained settings. The code is publicly available to facilitate reproduction and extension, and we hope it will serve as a foundation for further advances in ensemble methods for NLP.

References

[1] B. Pang and L. Lee, "Opinion mining and sentiment analysis," *Foundations and trends in information retrieval*, vol. 2, no. 1-2, pp. 1–135, 2008.

[2] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, "Learning word vectors for sentiment analysis," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2011, pp. 142–150.

[3] S. Wang and C. D. Manning, "Baselines and bigrams: Simple, good sentiment and topic classification," in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2*, 2012, pp. 90–94.

[4] R. Socher, J. Pennington, E. H. Huang, A. Y. Ng, and C. D. Manning, "Semi-supervised recursive autoencoders for predicting sentiment distributions," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2011, pp. 151–161.

[5] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[6] R. Kneser and H. Ney, "Improved backing-off for m-gram language modeling," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 1, 1995, pp. 181–184.

[7] T. Mikolov, M. Karafiát, L. Burget, J. Černockỳ, and S. Khudanpur, "Recurrent neural network based language model," in *Proceedings of the International Speech Communication Association*, 2010, pp. 1045–1048.

[8] T. G. Dietterich, "Ensemble methods in machine learning," *International Workshop on Multiple Classifier Systems*, pp. 1–15, 2000.

[9] A. Stolcke *et al.*, "Srilm-an extensible language modeling toolkit," in *Proceedings of the International Conference on Spoken Language Processing*, 2002, pp. 257–286.

[10] Q. V. Le and T. Mikolov, "Distributed representations of sentences and documents," in *International Conference on Machine Learning*, 2014, pp. 1188–1196.

[11] J. Platt *et al.*, "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods," *Advances in Large Margin Classifiers*, vol. 10, no. 3, pp. 61–74, 1999.

[12] L. I. Kuncheva and C. J. Whitaker, "That elusive diversity in classifier ensembles," *International Conference on Pattern Recognition and Image Analysis*, pp. 1126–1138, 2003.