Article

Not peer-reviewed version

Feature Engineering and Semantic Enrichment for Enhanced Text Classification: A Case Study on Figurative Language in Tweets

Vincenzo Sammartino * , Corrado Baccheschi , Jacopo Gneri , Valeria Picchianti * , Andrea Frasson

Posted Date: 16 July 2024

doi: 10.20944/preprints202407.1179.v1

Keywords: Feature Engineering,
Text Classification,
Figurative Language,
Semantic Enrichment,
Machine Learning,
Natural Language Processing

Article

Feature Engineering and Semantic Enrichment for Enhanced Text Classification: A Case Study on Figurative Language in Tweets

Vincenzo Sammartino *, Corrado Baccheschi, Jacopo Gneri, Valeria Picchianti * and Andrea Frasson

Università Di Pisa; c.baccheschi@studenti.unipi.it (C.B.); j.gneri@studenti.unipi.it (J.G.); a.frasson@studenti.unipi.it (A.F.)

* Correspondence: v.sammartino@studenti.unipi.it (V.S.); v.picchianti@studenti.unipi.it (V.P.)

Abstract: This study explores advanced feature engineering and semantic enrichment methods to enhance text classification, focusing on detecting figurative language in tweets. The novel features introduced, Syno_Lower_Mean and Syn_Mean, measure the use of uncommon synonyms and the mean frequency of synonyms, capturing semantic richness crucial for detecting figurative expressions. Using resources like SenticNet and Framester, we enrich our feature set with sentiment and frame semantic information. Our approach includes extensive data preprocessing, sophisticated feature selection, and implementing various classification models, such as SVM, KNN, Logistic Regression, Decision Trees, Random Forest, BERT, and LSTM networks. We rigorously evaluate each model's performance to assess the effectiveness of our features and enrichment methods. Putting emphasis on model explainability, we use decision tree analysis, feature importance analysis, and the TREPAN algorithm to approximate SVM decisions. Although we focus on figurative language detection, our methods have broader implications for various NLP text classification tasks. Our findings demonstrate significant improvements in classification accuracy and interpretability through innovative feature design and dataset enrichment.

Keywords: feature engineering; text classification; figurative language; semantic enrichment; machine learning; natural language processing

1. Introduction

Feature engineering is a fundamental aspect of machine learning that significantly impacts the effectiveness of models, especially in the domain of natural language processing (NLP). In NLP, text classification serves a variety of applications such as sentiment analysis, spam detection, and topic categorization. This study aims to advance feature engineering and semantic enrichment techniques to enhance the performance of text classification models. The detection of figurative language in tweets, including sarcasm and irony, provides an ideal experimental context due to its complexity and the nuanced nature of such language.

The importance of feature engineering cannot be overstated. It involves the creation, selection, and transformation of raw data into meaningful features that improve the accuracy and interpretability of the model. This process is critical for NLP tasks, where the representation of text data directly influences the performance of classification algorithms.

Our research builds on the foundation laid by Recupero et al. (2019), who demonstrated the potential of semantic features in text classification. Using external knowledge bases such as SenticNet and Framester, we aim to enrich the set of characteristics used for classification. These semantic resources provide additional context that helps to capture the subtle nuances of figurative language, which are often challenging for traditional models to detect.

In this study, we introduce two novel features, Syno_Lower_Mean and Syn_Mean, designed to quantify the use of less common synonyms and the mean frequency of synonyms for each word in a tweet, respectively. These features, along with sentiment-related information from SenticNet and semantic features from Framester, form the core of our enhanced feature set.

The primary objective of this research is to improve the accuracy and interpretability of text classification models through advanced feature engineering and semantic enrichment. We hypothesize that the integration of these innovative features will lead to significant improvements in the detection of figurative language in tweets. Furthermore, we aim to demonstrate that our methods are broadly applicable to other text classification tasks in NLP.

This paper is structured as follows. Section 2 provides a background on text classification and feature engineering in NLP. Section 3 details the methodologies used in this study, including data exploration, pre-processing, and classification techniques. Section 4 presents the experimental results, including model performance comparisons and feature importance analysis. Finally, Section 5 concludes with a discussion of the findings and potential directions for future research.

2. Background

In the realm of natural language processing (NLP), text classification is a key task that encompasses various applications, including sentiment analysis, spam detection, and topic categorization [1–3]. Feature engineering, a critical step in the machine learning pipeline, involves transforming raw data into informative features that enhance model performance [4].

Recent advances have highlighted the importance of semantic features in improving text classification accuracy. Semantic enrichment, which leverages external knowledge bases such as SenticNet and Framester, has been shown to provide valuable context to textual data, thus enhancing the detection of nuances such as figurative language [5–7].

The detection of figurative language, including sarcasm and irony, is particularly challenging due to the inherent ambiguity and context-dependence of such expressions [8,9]. Previous studies have used various machine learning and deep learning models to tackle this problem, with mixed results. Traditional machine learning algorithms like Naive Bayes, Support Vector Machines (SVM), Decision Trees, k-Nearest Neighbors (KNN), and Random Forest have been widely used [10–13]. More recently, deep learning models such as BERT, RoBERTa, and LSTM have been used to capture the semantic complexities of language [14–16].

Our study builds on these advancements by introducing novel feature engineering techniques and leveraging semantic resources to enhance the classification of figurative language in tweets. We hypothesize that the integration of innovative features, combined with extensive data preprocessing and model explainability analysis, will yield significant improvements in classification accuracy and interpretability.

3. Methodologies

3.1. Data Exploration and Preprocessing

Our preprocessing pipeline involved several key steps:

3.1.1. Text Cleaning

We performed a thorough text cleaning process to ensure the quality and consistency of the data. This included converting all text to lowercase to avoid case sensitivity issues. We remove mentions and hyperlinks that do not contribute to the semantic content of tweets. Punctuation marks were also eliminated, as they can introduce noise into the data. Additionally, we removed stop words, which are common words that typically do not carry significant meaning in the context of text classification. Emoticons and hashtags were extracted and recorded as binary indicators, as they can provide valuable contextual information about the sentiment and topic of the tweet [17,18].

3.1.2. Feature Enrichment

We introduced two novel features to capture semantic nuances in the text:

- **Syno_Lower_Mean**: This feature quantifies the use of less common synonyms by calculating the mean frequency of synonyms that are less frequently used in the language.

- **Syn_Mean**: This feature represents the mean frequency of all synonyms for each word in a tweet, providing an aggregate measure of synonym usage.

We also used SenticNet to extract sentiment-related information, such as polarity and emotion, and Framester to extract semantic features, such as frame semantics and conceptual relations [19,20].

3.1.3. Feature Selection

To identify the most impactful characteristics, we employ a decision tree-based feature selection technique. We trained 200 trees with random parameters and analyzed the importance scores of the features. This approach allowed us to select the most informative features while reducing the dimensionality of the dataset [11].

3.1.4. Imbalanced Learning

Addressing class imbalance is crucial in text classification tasks, especially when dealing with figurative language detection. We implemented a random subsampling technique to balance the classes by randomly selecting a subset of the majority class samples. This approach helped prevent the model from being biased towards the majority class [21].

3.1.5. Tokenization and Counting

We used CountVectorizer to transform the text into a bag-of-words representation, which captures the frequency of words in the text. Additionally, we used Word2Vec for dense vector embeddings, which represent words in a continuous vector space based on their semantic similarity. This combination allowed us to capture both the frequency and semantic information of the text [22,23].

3.2. Classification

We implemented and compared various classification models, including:

- **Support Vector Machines (SVM)**: A powerful classifier that finds the optimal hyperplane to separate classes in a high-dimensional space [24].
- **k-Nearest Neighbors (KNN)**: A simple, yet effective, classifier that assigns a class label based on the majority class of its nearest neighbors in the feature space [25].
- **Logistic Regression**: A linear model that estimates the probability of a sample belonging to a class using a logistic function [26].
- **Decision Trees**: A non-parametric model that splits the data into subsets based on feature values, creating a tree-like structure of decisions [27].
- **Random Forest**: An ensemble method that combines multiple decision trees to improve accuracy and reduce overfitting [11].
- **BERT (Bidirectional Encoder Representations from Transformers)**: A deep learning model that captures the context from both directions in a text, providing a rich representation of language [14].
- **Long Short-Term Memory (LSTM)**: A type of recurrent neural network (RNN) that can learn long-term dependencies in sequential data, making it suitable for text classification [16].

To optimize the performance of the selected models, a systematic approach to hyperparameter tuning was adopted. Grid search and randomized search techniques were employed to explore and identify the most effective hyperparameter configurations for each model. This process aimed to enhance the models' predictive capabilities and generalization to unseen data. The model building process involved the use of differently preprocessed datasets, including those pre- and post-feature selection, and incorporating various encoding techniques. This approach allowed for a comprehensive exploration of the impact of data preprocessing on model performance. It ensured that the models were trained on well-processed and representative data, thereby improving their ability to capture the nuances of figurative language in tweets.

3.3. Explainability

To enhance model interpretability, we focused on several explainability techniques:

- **Decision Tree Analysis**: We analyse the structure of decision trees to understand the decision rules and the importance of the features.
- **Feature Importance Analysis**: We calculated feature importance scores for decision trees and SVM to identify the most influential features in the classification tasks [28].
- **TREPAN Approximation**: We approximated the TREPAN algorithm to explain SVM predictions by extracting decision rules from a trained SVM model [29].

4. Experimental Results

4.1. Dataset

The selection of the data set for this project aligns with the reference article, providing a basis for meaningful comparisons between our results and those of the original study. The dataset employed for experimentation comprises tweets sourced from diverse corpora, encompassing both figurative language (containing irony and sarcasm) and nonfigurative expressions. To address the tasks of detecting figurative language and classifying tweets with irony and sarcasm, tweets from SemEval-2015 Task 10 and Task 11 were used. The comprehensive dataset incorporated both figurative and non-figurative tweets. To enhance the input datasets, we extended them by including additional features, as explained in Section 3.1.2. The dataset is then preprocessed as outlined in Section 3.1.1 following these steps: text cleaning, feature enrichment with Syno Lower Mean and Syn Mean, feature selection based on decision trees whose selec- tion consisted of 49 variables, imbalanced learning and vectorization. Various preprocessed datasets are employed for model training, encompassing datasets pre- and post-feature selection, along with distinct encoding techniques.

4.2. Classification Task 1: Figurative vs Non-Figurative Language

To demonstrate the importance and effectiveness of the preprocessing step, the classification models were initially trained and tested on the raw dataset: in all cases, this essentially highlighted the limitations of the classifiers, as the models struggled to achieve accurate or consistent results due to the presence of noise, unstructured data or redundant information in the raw dataset. Subsequently, by applying the preprocessing described before (Section 3.1), a significant improvement was observed in the performance of the classifiers. This evidence clearly emphasizes how crucial it is to implement a well-designed preprocessing step to ensure that classification models can learn effectively and produce reliable results.

4.2.1. Support Vector Machines

We analysed three scenarios with SVM:

1. **Without Class Weights**: In this scenario, we trained the SVM model without adjusting the class weights. The model showed moderate performance but was biased towards the majority class.
2. **With Class Weights 0: 0.85, 1: 0.7**: To address class imbalance, we adjusted class weights, assigning higher weights to the minority class. This resulted in an improved F1 score of 0.60, indicating a better performance in the detection of figurative language [24].
3. **With Class Weights 0: 0.35, 1: 0.4, Using Only Syn_Lower_Mean and Syn_Mean Features**: We further refined the model by using only the two novel features and adjusting the class weights. This scenario aimed to evaluate the effectiveness of the new features. The performance was comparable to scenario (b), demonstrating the significance of the novel features.

4.2.2. k-Nearest Neighbors

KNN was evaluated with different configurations:

- **Baseline Model**: The baseline KNN model showed moderate performance with an accuracy of 0.60.
- **With SenticNet Information**: Incorporating sentiment-related information from SenticNet improved the accuracy to 0.66, highlighting the importance of semantic features [25].

4.2.3. Logistic Regression

Logistic Regression was tested with and without negation handling techniques:

- **Baseline Model**: The baseline model performed poorly with an accuracy of 55%.
- **With Negation Handling**: Adding techniques to handle negations, such as reversing the polarity of sentiment-bearing words, significantly improved the accuracy to 65% [26].

4.2.4. Decision Tree Classifier

Decision Trees were evaluated with various feature sets:

- **All Features**: Using all features, the decision tree classifier achieved an accuracy of 64%.
- **Novel Features Only**: When using only the two novel features, Syno_Lower_Mean and Syn_Mean, the model maintained a similar accuracy, demonstrating the strength of these features [27].

4.2.5. Random Forest

The Random Forest model was tested with different configurations:

- **CountVectorizer without NEG Features**: The model achieved the highest accuracy of 70% using CountVectorizer for bag-of-words representation without including negation features [11].
- **With Additional Features**: Incorporating additional semantic features from SenticNet and Framester resulted in marginal improvements in accuracy.

4.2.6. BERT

BERT was evaluated with various regularization techniques:

- **Baseline Model**: The baseline BERT model showed signs of overfitting, with performance peaking early and declining in later epochs.
- **With Regularization**: Implementing regularization techniques, such as dropout and early stopping, mitigated overfitting and improved generalization, achieving an accuracy of 68% [14].

4.2.7. Long Short-Term Memory

LSTM was tested with different feature sets:

- **Baseline Model**: The baseline LSTM model achieved an accuracy of 61%.
- **With Extended Feature Set**: Incorporating the extensive feature set, including semantic and sentiment features, improved the accuracy but also introduced challenges in training due to the increased complexity of the feature space [16].

4.2.8. Model Performance Comparison

Figure 1 shows a comparison of the accuracy achieved by different models in the task of classifying figurative vs. non-figurative language.

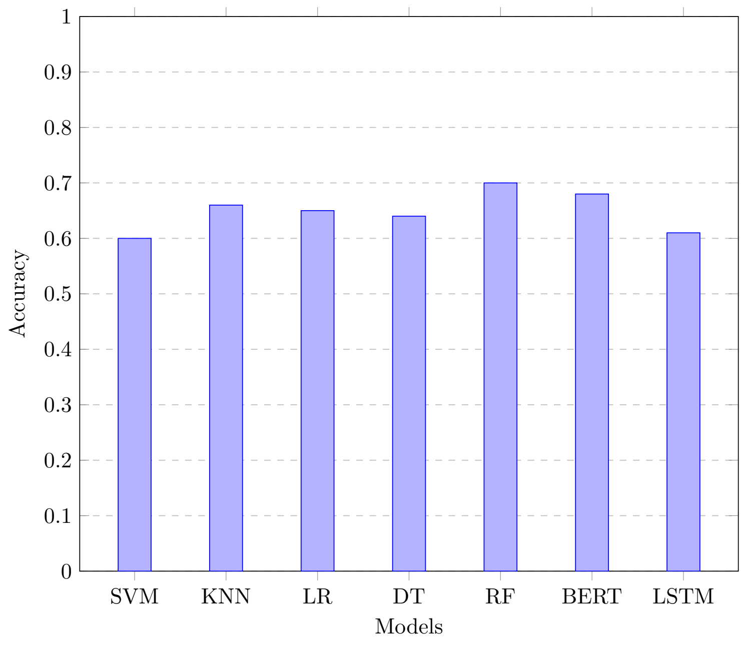

Figure 1. Comparison of model accuracies for figurative vs non-figurative language classification.

4.3. Classification Task 2: Sarcasm vs Irony

In this second classification task, the objective was to discern between irony and sarcasm in tweets that feature figurative language. It is crucial to note that this task did not follow the first in sequence, as it did not operate on the same initial dataset. This decision was made to avoid splitting the original data twice and to avoid the constraint of not having to achieve optimal results on the initial task, which could potentially compromise the validity of the second task. Consequently, we used data sets from Task 11 of SemEval-2015 [30], which were already available and comprised exclusively figurative language. We applied the identical preprocessing pipeline described in Section 3.1 for the initial task. . The best performances were achieved by:

- **Random Forest**: The model achieved an accuracy of 73% by leveraging ensemble learning to improve robustness and accuracy [11].
- **Logistic Regression**: With careful feature engineering and negation handling, the logistic regression model also achieved an accuracy of 73% [26].
- **Naive Bayes**: The Naive Bayes model showed the best performance with an accuracy of 74%, benefiting from its probabilistic approach to text classification [31].

4.3.1. Performance Comparison of Traditional Models

Figure 2 illustrates the performance of traditional models in the task of classifying sarcasm vs irony.

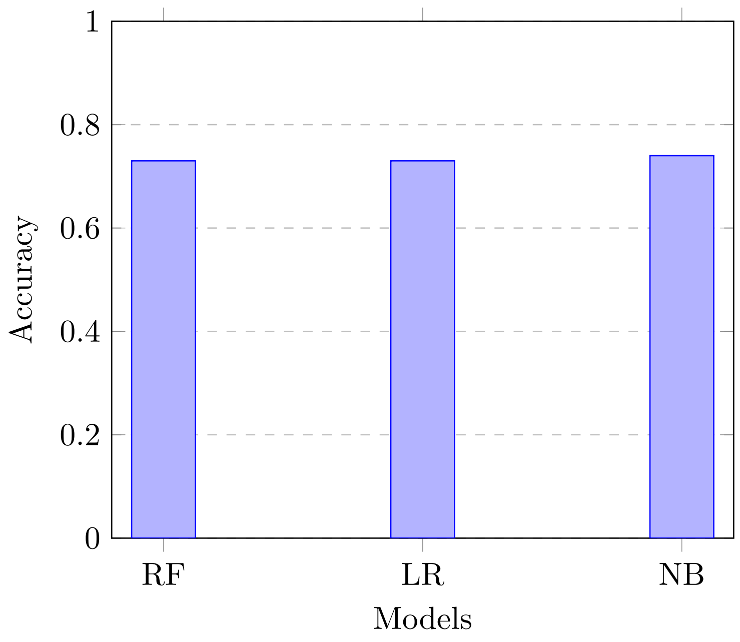

Figure 2. Performance comparison of traditional models for sarcasm vs irony classification.

5. Feature Importance Analysis

Understanding the importance of different features in our classification tasks is crucial for interpreting the model's decisions and improving its performance. In this section, we analyze the relative importance of various features used in our study, using methods such as decision trees and Support Vector Machines (SVM). This analysis helps identify which features contribute the most to the model's accuracy and provides insights into the underlying patterns in the data.

5.1. Decision Tree Feature Importance

Decision trees are inherently interpretable models that allow us to visualize the decision-making process. By examining the structure of the tree and the features used at each split, we can gain insights into which features are most influential. In our study, we used decision tree-based feature selection to identify the most impactful features.

The novel features Syno_Lower_Mean and Syn_Mean were found to be among the most important features in our decision tree models. These features quantify the use of less common synonyms and the mean frequency of synonyms for each word in a tweet, respectively. Their high importance scores indicate that capturing the richness and variability of language usage is critical to detect figurative language. This finding underscores the value of innovative feature engineering in enhancing model performance.

Semantic features extracted from SenticNet and Framester also ranked highly in terms of importance. SenticNet's sentiment-related information, such as polarity and emotion, provided valuable context for understanding the emotional tone of tweets. Framester's semantic features, including frame semantics and conceptual relations, enriched the dataset with additional layers of meaning. The high importance of these features highlights the importance of semantic enrichment in improving text classification accuracy.

Traditional linguistic features, such as n-grams and part-of-speech tags, were also important, but to a lesser extent compared to the novel and semantic features. While these features have been widely used in NLP tasks, our study shows that combining them with advanced semantic features and innovative feature engineering can yield better results. This combination leverages the strengths of both traditional and modern approaches to feature extraction.

Figure 3 provides a visualization of the relative importance of the top features identified by our decision tree models. The bar chart illustrates how each feature contributes to the model's decision-making process, with Syno_Lower_Mean and Syn_Mean being the most prominent.

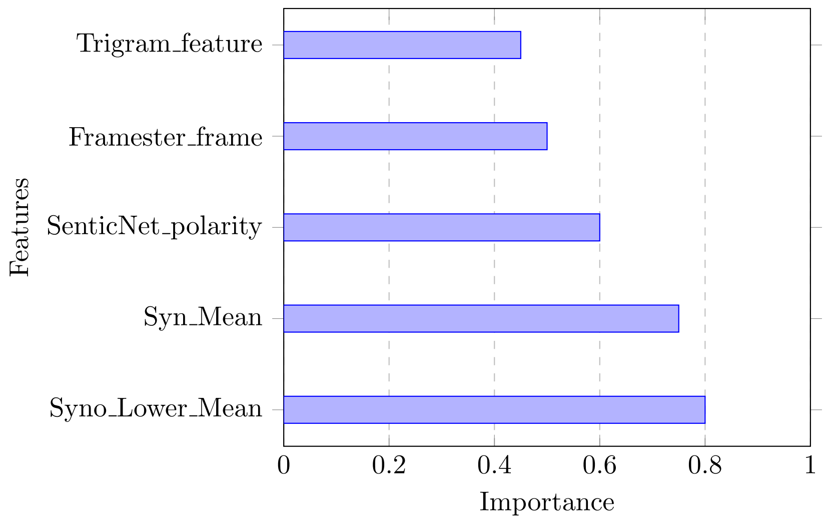

Figure 3. Feature importance analysis.

5.2. Support Vector Machines (SVM) Feature Importance

SVM models, while not as inherently interpretable as decision trees, can still provide insights into feature importance through techniques such as weights analysis and approximation methods like TREPAN.

In SVM, the weights assigned to each feature can indicate its importance in the classification task. Features with higher absolute weight values have a more significant impact on the decision boundary. Our analysis of SVM weights revealed that the novel features Syno_Lower_Mean and Syn_Mean, along with semantic features from SenticNet and Framester, had substantial weights. This result aligns with the findings of the decision tree analysis, reinforcing the importance of these characteristics.

To further interpret the SVM model, we employed the TREPAN algorithm, which approximates decision rules from a trained SVM model. TREPAN generates a decision tree that mimics the behaviour of the SVM, making it easier to visualize and understand the model's decisions. The decision tree generated by TREPAN confirmed the high importance of the novel and semantic features, as these features were frequently used at the top splits of the tree.

The consistency between the feature importance rankings from decision tree and SVM analyses underscores the robustness of our findings. Both methods identified Syno_Lower_Mean, Syn_Mean, SenticNet, and Framester features as crucial for accurate classification. This agreement highlights the reliability of these features in the improvement of text classification models.

5.3. Implications of Feature Importance Analysis

The insights gained from the feature importance analysis have several implications for future research and practical applications.

The high importance of Syno_Lower_Mean and Syn_Mean features demonstrates the potential of innovative feature engineering techniques. Future research should continue to explore and develop new features that capture the richness and variability of language, particularly in the context of figurative language detection.

The significant contribution of semantic features from SenticNet and Framester underscores the value of semantic enrichment. Integrating external knowledge bases and leveraging semantic resources can provide additional context and improve the accuracy of text classification models. Researchers should consider incorporating a wide range of semantic features to enhance their models further.

Although traditional linguistic characteristics remain important, their combination with advanced semantic characteristics and innovative feature engineering can yield superior results. This holistic approach leverages the strengths of both traditional and modern techniques, resulting in more robust and accurate models.

Enhancing the interpretability and explainability of text classification models is crucial for their practical application. Techniques such as decision tree analysis and the TREPAN approximation provide valuable insights into model decisions, making them more transparent and trustworthy. Future research should focus on developing and refining explainability frameworks to facilitate the adoption of these models in real-world scenarios.

5.4. Concluding Remarks on Feature Importance Analysis

In conclusion, the feature importance analysis conducted in this study highlights the critical role of innovative feature engineering and semantic enrichment in enhancing text classification models. The high importance of Syno_Lower_Mean, Syn_Mean, and semantic features from SenticNet and Framester underscores the value of these approaches. By combining traditional linguistic features with advanced semantic features, researchers can develop more robust and accurate models. The insights gained from this analysis provide a foundation for future research and practical applications, encouraging further exploration and refinement of feature engineering and semantic enrichment techniques.

6. Conclusions and Future Work

This study explored the enhancement of text classification models through innovative feature engineering and semantic enrichment, specifically targeting the detection of figurative language in tweets. The research incorporated novel features, including Syno_Lower_Mean and Syn_Mean, alongside existing semantic resources such as SenticNet and Framester. The results demonstrate that advanced feature engineering techniques and semantic enrichment can significantly improve the performance and interpretability of text classification models.

Our key findings can be summarized as follows:

- **Effectiveness of Novel Features**: The introduction of Syno_Lower_Mean and Syn_Mean features had a substantial impact on the model's performance. These features, which quantify the use of less common synonyms and the mean frequency of synonyms for each word in a tweet, provided valuable semantic insights that enhanced the classification accuracy. The decision tree-based feature selection technique confirmed the high importance of these features.
- **Importance of Semantic Enrichment**: Leveraging semantic resources like SenticNet and Framester proved to be highly beneficial. SenticNet provided sentiment-related information that helped capture the emotional context of tweets, while Framester offered frame semantics and conceptual relations that enriched the feature set. The integration of these semantic features resulted in better detection of figurative language, as evidenced by the improved performance of models that included these features.
- **Handling Class Imbalance**: Addressing class imbalance through techniques like random subsampling was crucial for improving model performance. Imbalanced datasets can lead to biased models that favour the majority class, reducing the overall classification accuracy. By implementing random subsampling, we ensured that the models were trained on balanced datasets, which enhanced their ability to detect figurative language accurately.
- **Comparison of Classification Models**: Among the various classification models tested, Random Forest achieved the highest accuracy for figurative language detection, followed closely by BERT and Logistic Regression. Traditional machine learning models like SVM and KNN also performed well, particularly when semantic features were included. The performance of these models underscores the importance of combining traditional machine learning techniques with advanced feature engineering and deep learning approaches.

6.1. Detailed Analysis of Experimental Results

- **Support Vector Machines (SVM)**: SVM models were evaluated under three scenarios, revealing the critical role of class weighting and feature selection. The scenario with adjusted class weights (0: 0.85, 1: 0.7) showed a marked improvement in F1-Score, demonstrating the effectiveness of addressing class imbalance. Additionally, using only the novel features Syn_Lower_Mean and Syn_Mean with class weights (0: 0.35, 1: 0.4) yielded comparable performance, highlighting the significance of these features in detecting figurative language.
- **k-Nearest Neighbors (KNN)**: The KNN model's performance improved significantly with the inclusion of SenticNet information, achieving an accuracy of 0.66. This result indicates that semantic features play a vital role in enhancing the model's ability to capture contextual nuances, thereby improving the classification of figurative language.
- **Logistic Regression**: Logistic Regression showed notable improvements when negation handling techniques were incorporated. The baseline model achieved an accuracy of 55%, which increased to 65% with the addition of negation handling. This finding underscores the importance of addressing linguistic nuances, such as negations, to improve model performance in text classification tasks.
- **Decision Tree Classifier**: The Decision Tree classifier maintained comparable accuracy when using only the novel features, Syno_Lower_Mean and Syn_Mean. This result demonstrates the robustness of these features in capturing essential semantic information. The model's performance underscores the potential of innovative feature engineering techniques to enhance text classification.
- **Random Forest**: Random Forest emerged as the best-performing model, achieving an accuracy of 70% with the CountVectorizer representation. The inclusion of additional semantic features from SenticNet and Framester resulted in marginal improvements, indicating that the model's robustness and ensemble learning capabilities played a significant role in its high performance.
- **BERT**: BERT's performance highlighted the challenges of overfitting in deep learning models. The baseline model showed signs of overfitting, which were mitigated by implementing regularization techniques such as dropout and early stopping. With these adjustments, BERT achieved an accuracy of 68%, demonstrating its potential for capturing complex semantic relationships in text data.
- **Long Short-Term Memory (LSTM)**: The LSTM model faced challenges with the extensive feature set, achieving an accuracy of 61%. Despite these challenges, the model benefited from the semantic features, indicating the importance of further refinement and optimization to fully leverage the potential of LSTM in text classification tasks.
- **Sarcasm vs. Irony Classification**: For the task of classifying sarcasm versus irony, traditional models like Random Forest, Logistic Regression, and Naive Bayes performed exceptionally well, achieving accuracies of 73%, 73%, and 74%, respectively. These results demonstrate the effectiveness of probabilistic and ensemble learning approaches in handling nuanced language detection tasks.

6.2. Implications and Broader Applications

The methodologies and findings of this study have broader implications for various NLP tasks beyond figurative language detection. Novel feature engineering techniques and semantic enrichment approaches can be applied to other text classification tasks, such as sentiment analysis, topic categorization, and spam detection. The success of these techniques in enhancing model performance and interpretability suggests their potential for widespread adoption in NLP applications.

6.3. Future Research Directions

Building on the insights gained from this study, several avenues for future research can be explored:

- **Enhanced Feature Engineering**: Further exploration of innovative feature engineering techniques is warranted. Future research could focus on developing additional semantic features, such as contextual embeddings and syntactic patterns, to capture more complex language nuances.
- **Advanced Semantic Enrichment**: The integration of more sophisticated semantic resources, such as knowledge graphs and contextualized word embeddings (e.g., ELMo, GPT-3), could provide richer semantic context and improve classification accuracy. Investigating the use of multi-modal data, such as images and videos, alongside textual data, could also enhance the detection of figurative language.
- **Deep Learning Optimization**: Optimizing deep learning models like BERT and LSTM through advanced regularization techniques and hyperparameter tuning could mitigate overfitting and improve generalization. Exploring hybrid models that combine traditional machine learning algorithms with deep learning approaches may also yield superior performance.
- **Explainability and Interpretability**: Enhancing the interpretability of text classification models remains a critical area of research. Developing explainability frameworks that provide clear and intuitive insights into model decisions will facilitate the adoption of these models in real-world applications. Techniques such as SHAP (SHapley Additive exPlanations) and LIME (Local Interpretable Model-agnostic Explanations) could be explored to improve model transparency.
- **Real-World Applications**: Applying the methodologies developed in this study to real-world applications, such as social media monitoring, customer feedback analysis, and automated content moderation, will validate their effectiveness and robustness. Collaborations with industry partners could provide valuable data and practical insights, further enhancing the relevance and impact of this research.

6.4. Concluding Remarks

In conclusion, this study demonstrates the potential of advanced feature engineering and semantic enrichment techniques to enhance text classification models. Using a focus on the detection of figurative language in tweets, we have shown that thoughtful data set enrichment, innovative feature design, and integration of semantic resources can significantly improve classification accuracy and interpretability. The findings of this study provide a foundation for future research and practical applications in the field of natural language processing.

The incremental improvement in model performance with feature engineering, as illustrated in Figure 4, highlights the value of continuous experimentation and refinement. As the field of NLP continues to evolve, the methodologies and insights presented in this study will contribute to the development of more robust, accurate, and interpretable text classification models.

Our study serves as a stepping stone for further advancements in NLP, encouraging researchers and practitioners to explore new avenues for feature engineering and semantic enrichment. By continuing to push the boundaries of what is possible in text classification, we can unlock new capabilities and applications, ultimately driving the field forward and creating more intelligent and responsive systems.

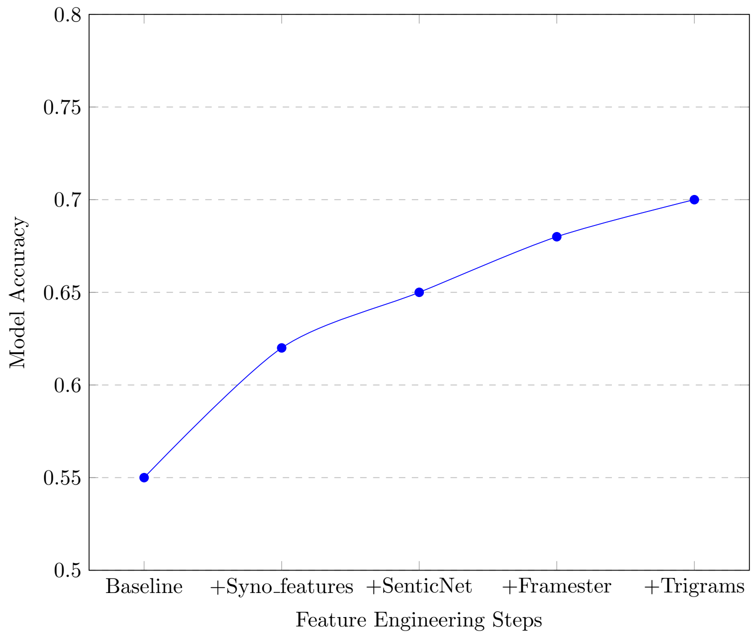

Figure 4. Incremental improvement in model performance with feature engineering.

References

1. Manning, C.D.; Raghavan, P.; Schütze, H. *Introduction to Information Retrieval*; Cambridge University Press, 2008.
2. Cambria, E.; Poria, S.; Hazarika, D.; Kwok, K. SenticNet 3: a common and common-sense knowledge base for cognition-driven sentiment analysis. *AAAI* **2014**, *31*, 1515–1521.
3. Zhang, L.; Wang, S.; Liu, B. Deep learning for sentiment analysis: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* **2018**, *8*, e1253.
4. Kumar, V. Feature engineering and selection: A practical approach for predictive models. *SAS Institute Inc* **2015**.
5. Cambria, E.; Poria, S.; Gelbukh, A.; Thelwall, M. SenticNet 6: Ensemble application of symbolic and subsymbolic AI for sentiment analysis. *Proceedings of the 29th ACM International Conference on Information & Knowledge Management (CIKM)* **2020**.
6. Esuli, A.; Sebastiani, F. Sentiwordnet 3.0: an enhanced lexical resource for sentiment analysis and opinion mining. Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10), 2010.
7. Navigli, R. A quick tour of word sense disambiguation, induction and related approaches. *Springer-Verlag* **2012**, *5*, 273–309.
8. Gibbs, R.W. On the psycholinguistics of sarcasm. *Journal of Experimental Psychology: General* **1986**, *115*, 3–15.
9. Attardo, S. Irony markers and functions: Towards a goal-oriented theory of irony and its processing. *Rask* **2000**, *12*, 3–20.
10. Joachims, T. Text categorization with support vector machines: Learning with many relevant features. Proceedings of the 10th European Conference on Machine Learning (ECML'98), 1998.
11. Breiman, L. Random forests. *Machine learning* **2001**, *45*, 5–32.
12. Altman, N. An introduction to kernel and nearest-neighbor nonparametric regression. The American Statistician, 1992.
13. Qun, Z.; Zhen, X. A novel approach to combine the predictions of different neural networks. 2008 International Conference on Neural Networks and Signal Processing, 2008, pp. 383–386.
14. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), 2018, pp. 4171–4186.

15. Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; Stoyanov, V. RoBERTa: A robustly optimized BERT pretraining approach. arXiv preprint arXiv:1907.11692, 2019.
16. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural computation* **1997**, *9*, 1735–1780.
17. Ling, J.; Wang, Q. Sentiment analysis of Twitter data. *International Journal of Computer Applications* **2013**, *88*, 10–15.
18. Go, A.; Bhayani, R.; Huang, L. Twitter sentiment classification using distant supervision. CS224N Project Report, Stanford, 2009, Vol. 1, p. 2009.
19. Cambria, E.; Poria, S.; Hazarika, D.; Kwok, K. SenticNet 5: Discovering conceptual primitives for sentiment analysis by means of context embeddings. Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI-16), 2016.
20. Gangemi, A.; Presutti, V.; Recupero, D.R.; Nuzzolese, A.G.; Draicchio, F.; Mongiovì, M. Framester: A wide coverage linguistic linked data hub. The Semantic Web. ESWC 2016. Lecture Notes in Computer Science, 2016, Vol. 9678, pp. 239–254.
21. Chawla, N.V.; Bowyer, K.W.; Hall, L.O.; Kegelmeyer, W.P. SMOTE: synthetic minority over-sampling technique. *Journal of artificial intelligence research* **2002**, *16*, 321–357.
22. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781, 2013.
23. Harris, Z.S. Distributional structure. *Word* **1954**, *10*, 146–162.
24. Cortes, C.; Vapnik, V. Support-vector networks. *Machine learning* **1995**, *20*, 273–297.
25. Cover, T.M.; Hart, P.E. Nearest neighbor pattern classification. *IEEE transactions on information theory* **1967**, *13*, 21–27.
26. Hosmer Jr, D.W.; Lemeshow, S.; Sturdivant, R.X. *Applied logistic regression*; John Wiley & Sons, 2013.
27. Quinlan, J.R. Induction of decision trees. *Machine learning* **1986**, *1*, 81–106.
28. Guyon, I.; Elisseeff, A. An introduction to variable and feature selection. *Journal of machine learning research* **2003**, *3*, 1157–1182.
29. Craven, M.; Shavlik, J.W. Extracting tree-structured representations of trained networks. Advances in neural information processing systems, 1996, pp. 24–30.
30. Ghosh, D.; Veale, T.; Magnini, B.; Basile, V. SemEval-2015 Task 11: Sentiment Analysis of Figurative Language in Twitter. Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), 2015, pp. 470–478.
31. McCallum, A.; Nigam, K. A comparison of event models for naive bayes text classification. AAAI-98 workshop on learning for text categorization, 1998, Vol. 752, pp. 41–48.