Ingénierie des Systèmes d'Information
Vol. 30, No. 6, June, 2025, pp. 1469-1482
Journal homepage: http://iieta.org/journals/isi

Performance Evaluation of Text Embedding Models for Ambiguity Classification in Indonesian News Corpus: A Comparative Study of TF-IDF, Word2Vec, FastText BERT, and GPT

Sutriawan[1,2*], Supriadi Rustad[3], Guruh Fajar Shidik[4], Pujiono[1]

[1] Faculty of Computer Science, Universitas Dian Nuswantoro, Semarang 50131, Indonesia
[2] Department of Computer Science, Universitas Muhammadiyah Bima, Bima 84113, Indonesia
[3] Research Centre for Quantum Computing and Material Informatics, Faculty of Computer Science, Universitas Dian Nuswantoro, Semarang 50131, Indonesia
[4] Pusat Kajian Intelligent Distributed Surveillanced and Security, Universitas Dian Nuswantoro, Semarang 50131, Indonesia

Corresponding Author Email: p41202200046@mhs.dinus.ac.id

Copyright: ©2025 The authors. This article is published by IIETA and is licensed under the CC BY 4.0 license (http://creativecommons.org/licenses/by/4.0/).

https://doi.org/10.18280/isi.300606

Received: 14 May 2025
Revised: 16 June 2025
Accepted: 26 June 2025
Available online: 30 June 2025

Keywords:
text embedding, ambiguity detection, Indonesian NLP, TF-IDF, Word2Vec, FastText, BERT, GPT

ABSTRACT

Ambiguity in sentence classification is a major challenge in natural language processing (NLP), as it requires a deep understanding of complex semantic contexts. Although various text embedding models have been applied to text classification tasks, comprehensive evaluations of their effectiveness in detecting ambiguous sentences, particularly in Indonesian news corpora, remain limited. This study addresses that gap by comparing the performance of five text embedding models TF-IDF, Word2Vec, FastText, BERT, and GPT combined with five binary classification algorithms: Logistic Regression, Random Forest, bagging, Multinomial Naive Bayes, and Gaussian Naive Bayes. The dataset was derived from the XL-Sum Indonesian news corpus, with sentences automatically labeled as ambiguous or unambiguous using the Claude 3.5 language model. Experimental results show that the combination of Gaussian Naive Bayes with GPT embeddings achieved the best performance in ambiguous sentence classification, with a recall of 71% and an F1-score of 60%. Meanwhile, the combination of TF-IDF with bagging yielded the highest accuracy of 83% for unambiguous sentence classification. These findings highlight the critical role of selecting appropriate embedding and classification models to enhance accuracy in semantically ambiguous sentence classification for the Indonesian language.

1. INTRODUCTION

Textual data on online news platforms is growing rapidly, as more and more information is produced and disseminated through various digital channels [1]. A vast amount of varied data is produced daily by the millions of news stories, opinions, and articles that are uploaded. This presents both a difficulty and an opportunity for the field of natural language processing (NLP), especially when it comes to organizing, categorizing, and gleaning pertinent data from the ever-expanding body of text [2-4]. As a result of the increasing amount of content available, it is important to handle a variety of information, including that which is ambiguous or unclear. Ambiguities in text often arise due to different interpretations of words or sentences, which can be influenced by social context, culture, or even language changes over time. For example, words or phrases in news articles can have more than one meaning, depending on how and where they are used [5].

Ambiguity is an inherent characteristic of human language, where a single word or sentence can have multiple interpretations based on context, word meaning, or sentence structure. There are many difficulties with this language issue, particularly when it comes to classifying news texts. Ambiguity becomes a significant issue in the area of text categorization when attempting to assign a text the appropriate label. For instance, *I will meet you at the bank tomorrow* can signify different things to different people. Depending on the proper interpretation, the term *bank* can refer to either a financial institution or the riverbank, resulting in several categories. An automated system could easily misclassify the sentence if it doesn't have a good knowledge of the context [5-7].

In the context of ambiguous sentence classification, the main focus lies on identifying and distinguishing between sentences that have more than one meaning or interpretation based on their context. This process involves the use of models designed to distinguish between ambiguous and unambiguous sentences. The goal of classification is to determine if a sentence has more than one valid interpretation or if the meaning is clear. The approach used is binary classification, where sentences are grouped into two classes:

- **Ambiguous:** Sentences that have more than one meaning depending on the context.
- **Unambiguous:** Sentences whose meaning is clear and leave no doubt.

The problem of ambiguity in news classification is one of

the main challenges to overcome, especially since news often contains complex, multi-interpretive, or confusing information. Ambiguity in news can arise from various factors, such as insufficient context, unclear language, or different interpretations of an event. In addition, news often presents information spread across several interrelated sentences or paragraphs. Ambiguity can occur when the analysis does not consider the broader context. In many cases, the meaning of a sentence or phrase in the news can change or become unclear without considering the entire context. Therefore, an approach that takes into account the global context is crucial to overcoming this challenge.

Text ambiguity classification using machine learning has gained attention in various fields. An automated approach combining text mining and machine learning has been proposed in software engineering to detect ambiguous software requirement specifications, thus potentially reducing project risk. Ambiguous software requirement specifications can cause problems in software development [8]. For Support Vector Machine classifiers, feature selection techniques like the Ambiguity Measure can drastically cut down on training time without sacrificing accuracy. The issue occurs when classifying vast volumes of data becomes challenging due to the enormous dimensionality of the feature space in text classification [6]. A comparative analysis of various classification algorithms, including Naive Bayes, SVM, Random Forests, and Decision Trees, has been conducted using the Weka tool to identify ambiguities in requirements engineering documents [9, 10]. The problem in this research is the comparison of various supervised text classification algorithms on datasets with part-of-speech ambiguity. This research shows that Decision Tree often performs well in terms of accuracy and computational efficiency. In addition, a hybrid model that integrates multiple algorithms has been proposed to further improve the classification accuracy [9].

Current ambiguity handling methods still face challenges despite rapid advances in technology and models that have been developed, such as transformer-based models, deep learning, and machine learning. Ambiguous text detection by utilizing machine learning algorithms to detect ambiguous software requirements. In order to identify unclear software needs, nine distinct machine learning classification algorithms were evaluated [8]. Although it requires more time to train, the Support Vector Machine (SVM) classification algorithm outperforms other text classification algorithms. In order to address the high-dimensional challenge, the Ambiguity Measure (AM) feature selection method is required. It claims to cut training time by over 50% while preserving or enhancing text classifier accuracy when compared to using the entire feature set [6]. Several supervised text classification algorithms, including Naive Bayes, SVM, and Random Forests, are applied to a dataset of 2000 sentences with part-of-speech ambiguity. The accuracy, F-score, recall, and precision of the various classification algorithms are tested and evaluated using a confusion matrix, and the results show that each algorithm's accuracy is only 66–84%. The hybrid model AmbiF is suggested to raise the accuracy to 85% [9]. In automating ambiguity detection in software requirement specification (SRS) documents by using two text classification systems, the algorithm's performance is evaluated using two feature sets: one at the sentence level and one at the discourse level. The text classification system's matrix accuracy in identifying ambiguity is 86.67%. However, this study's shortcoming is that it solely examines the use of text classification to automate requirements engineering ambiguity detection; as a result, no prior research can be compared [11]. When employing machine learning methods for semantic disambiguation of ambiguous text data, the issue is decreased interpretability and accuracy. The suggested method or algorithm is a hybrid approach that combines the effectiveness of machine learning optimization with the ability to distinguish ambiguities as features that are both semantically and interpretively significant [12]. Three strategies are suggested in this study to enhance the effectiveness of semantic relatedness metrics. With four improvement methods—dimension removal, offset transformation, uniform transformation, and harmonic series transformation—the first approach tackles abnormal dimensions in word embedding models like GloVe and HPCA. It has been demonstrated that removing abnormal dimensions significantly improves performance. The second approach combined linear word embedding with WordNet path information, which successfully outperformed the comparison model on seven out of eight benchmark datasets. The third approach used support vector regression (SVR) to combine features from WordNet and word embedding, resulting in the best performance against all benchmark models. These findings show weaknesses in current optimization algorithms and highlight the potential of combining WordNet with word embedding. Future plans include further exploration of abnormal dimensions, broader utilization of WordNet information, and development of WordNet-based regression methods [13].

The contribution of this research presents a new approach in the classification of ambiguous sentences in Indonesian through a thorough evaluation of five main text embedding models (TF-IDF, Word2Vec, FastText, BERT, and GPT) integrated with binary classification algorithms. The main contributions of this study are (1) Development of Embedding Evaluation Benchmark for Ambiguity Detection in Indonesian, to date, there has been no comprehensive study comparing the performance of classical and transformer-based embedding directly for ambiguous sentence detection tasks in Indonesian. This research introduces the first experimental benchmark on the XL-Sum Indonesian news corpus, which involves more than 90,000 sentence data that have been labeled with ambiguities. (2) Innovation in Dataset Labeling Process with Large Language Model (Claude 3.5), To overcome the challenges of limited human annotators and subjectivity in determining ambiguity, this research utilizes Claude 3.5 as an automatic semantic classification model to label ambiguous or unambiguous. This approach is the first LLM-based large-scale annotation strategy applied to Indonesian data and can be replicated for other corpus in the future. (3) Combination of GPT Model with Gaussian Naive Bayes as a Contextual Classification Strategy, this study shows that the combination of GPT embedding with Gaussian Naive Bayes produces the best performance for detecting ambiguous sentences, especially in the recall metric (0.71) and F1-score (0.60) in the ambiguous class. These results show that simple probabilistic models such as GNB can achieve high performance when combined with rich contextual embedding. This proves that the low-complexity classifier + high-semantic embedding strategy can be an efficient solution for ambiguity detection, especially in computationally constrained environments. (4) Performance and Generalization Analysis of Embedding Variants, this research provides an in-depth understanding of how each embedding model works in handling ambiguous contexts. It is shown that frequency-based embedding (TF-

IDF) has high accuracy but is weak in ambiguous class recall, while models such as Word2Vec and FastText are less able to capture global semantic relations. In contrast, GPT and BERT showed superior ability in understanding semantic nuances and hidden context meanings.

The structure of this paper consists of Section 1, discussing the introduction, phenomena and problems raised in this research. Section 2 reviews relevant previous research and explains the position and differences of this research approach compared to previous studies. Section 3 presents the proposed method, including the data labeling process, text embedding modeling, and classification. Section 4 presents the experimental results and performance analysis of various model combinations. Finally, Section 5 summarizes the main findings and provides future research directions.

2. RELATED WORKS

The assessment of embedding models for Indonesian text in ambiguous sentence categorization using binary classifiers is a crucial topic of study in the field of text processing. The usefulness of various models and methods for text analysis tasks, particularly in the context of sentence classification, has been the subject of numerous research. There are issues with ambiguity in requirements engineering papers written in plain language, which can hinder the software development process and compromise the end product's quality. In order to automatically categorize requirements engineering papers as ambiguous or unambiguous at the syntactic level, a text classification technique is presented. The objective is to minimize or reduce ambiguity in requirements engineering papers by using machine learning approaches to identify its presence [6, 8, 10]. Using the WordNet database as a guide, this study suggests three approaches for determining the meaning of social tags: co-occurrence-based methods, Lin-based information theory, and latent semantic analysis (LSA). Findings indicate that the co-occurrence and LSA_w approaches perform best, with LIN and LSA_w being more resilient to changes in the quantity of tag meanings and Boolean effects. However, the highest decision success rate is achieved by the co-occurrence method. This research is limited to tag meanings defined in WordNet and data from Delicious, but the approach can be extended to other platforms such as Flickr and YouTube. Future plans include involving more tags in the experiments as well as investigating the connection between collective intelligence in social tagging and tag meaning disambiguation [14]. Tackle the problem of word meaning disambiguation by utilizing the technique of the word embedding model, which is to be trained with the RNN-LSTM model and then measured for the meaning value. The limitation is the unavailability of context. The purpose is to classify the meaning of ambiguous words and map the output to WordNet to retrieve the correct meaning. [7]. The necessity for efficient techniques to resolve word meaning ambiguity in biomedical texts is addressed in this study. Scholars employ supervised learning methods, including artificial neural networks based on Long Short-Term Memory (LSTM) and Support Vector Machine (SVM), which utilize features from the context of ambiguous words as well as global information extracted from MEDLINE through word embedding. The findings demonstrate that on the MSH WSD dataset, the combination of unigrams and SVM-based word embedding yields better results [15]. This research addresses the problem of ambiguity in software written using natural language, where ambiguous words can be interpreted differently by stakeholders from different domain expertise. The technique used is a word embedding-based natural language processing approach to detect and quantify the potential ambiguity of such words across different subdomains. This research evaluates the effectiveness of word embedding in identifying domain-specific ambiguities, with the primary objective of detecting and resolving ambiguous words in natural language text [16]. The issue of word meaning ambiguity in code-mixed social media texts—where word meanings might change according on the context—is the main topic of this study. Using a hierarchical Long Short-Term Memory (LSTM) model, the researcher suggests a character embedding-based method to determine the language and context of ambiguous words in phrases with mixed codes. The primary goal of this study is to use machine learning to clarify word meaning ambiguity in code-mixed text [17]. This research addresses the importance of addressing ambiguity in natural language, especially in information retrieval. Word embedding is able to capture semantic information but is less effective in handling ambiguity, especially on single-word queries. While transformer models can handle ambiguity in complex queries, their use is limited by high training costs and the need for sensitive data. As a more efficient solution, this study uses DBSCAN clustering on latent space to identify and evaluate word ambiguity. This method produces clusters that are semantically coherent and accurately reflect the meaning of words while offering a more resource-efficient approach than the transformer model [18]. The topic of text classification utilizing embedded words to represent the text is the main subject of this study. The objective is to assess how well text categorization performs using word embedding representation, which makes it possible to model semantic relationships in text more effectively [19]. This research addresses the problem that machines cannot inherently understand natural language meaning and require human parameters to model semantic relationships. By using local context information, the newly presented Content Tree Word Embedding (CTWE) technique seeks to address the shortcomings of current word embedding techniques. The primary goal of this study is to demonstrate that, in contrast to other word embedding techniques like GloVe and Word2Vec, the CTWE strategy can enhance performance in document categorization tasks [20]. This research addresses the problem of word meaning disambiguation in a clinical context, specifically for deciphering clinical abbreviations. The technique used involves training SVM and Naive Bayes models with four different strategies that integrate pre-trained word embeddings as features. The findings demonstrated that, when employing pre-trained models from Wikipedia, PubMed, and PMC texts, the SVM model had the maximum accuracy of 97.08%. However, this study has limitations related to the lack of resources for low-resource languages, which requires new approaches to extract abbreviations and their definitions from clinical narratives. This study's primary goal is to evaluate the effectiveness of two machine learning algorithms employing pre-trained word embeddings and various feature types [21]. Addresses the problem of acronym ambiguity in scientific papers by developing a method to decipher acronyms. The technique used compares the context vector of the acronym with the weighted average vector of the words in its expansion. Its performance is measured by comparing the results with the classical cosine similarity approach. The main objectives of

this research are to develop a widely applicable acronym parsing technique that does not require training for each acronym and to show that word embeddings trained on scientific texts perform better than word embeddings trained on a general corpus [22]. Transformers approaches such as the BERT Model have been extensively analyzed and evaluated in terms of lexical ambiguity, demonstrating their effectiveness in dealing with word sense disambiguity tasks [23]. The evaluation of embedding models, particularly BERT, for Indonesian text processing in ambiguous sentence classification using binary classifiers, is still an important area of research. The significance of insertion models in natural language processing has been demonstrated by the notable improvement in text classification task performance achieved through the combination of deep learning models, hybrid feature extraction approaches, and pre-trained language models [24]. Previous research in ambiguity detection has generally focused on English-language corpus, such as software specification documents or medical texts, with approaches such as Word Sense Disambiguation (WSD) [25, 26], rule-based methods, or simple ML models. However, such approaches are less adaptive to low-resource languages such as Bahasa Indonesia, especially in the context of news [27]. Moreover, there have not been many studies that explicitly compare traditional and transformer-based embedding for ambiguity detection tasks. This research fills the gap by systematically benchmarking text embedding on ambiguous/unambiguous labeled datasets in Bahasa Indonesia.

3. PROPOSED METHOD

This study aims to evaluate various embedding models in the classification of ambiguous sentences in the BBC Indonesia news corpus using binary classifiers. In this context, embedding approaches are crucial to semantically represent the text and capture the context of ambiguous sentences. The tested embedding model aims to improve classification accuracy by overcoming the challenges of language ambiguity. The method used in this research involves various steps, from data preprocessing to the application of machine learning models to classify sentences into binary categories. This approach is designed to measure the effectiveness of embedding in understanding the ambiguity and context of Indonesian in a news corpus. Figure 1 illustrates the design of this proposed study.

3.1 Indonesian news corpus dataset

The data used in this study were obtained from the hugging face which is the content of XL-Sum BBC News which can be accessed through the portal https://huggingface.co/datasets/csebuetnlp/xlsum/viewer/indonesian which consists of training, testing and validation data [28].

3.2 Labelling dataset

The ambiguity labeling process in this study was conducted automatically using Claude 3.5 Sonnet, a large language model (LLM) developed by Anthropic, which has demonstrated strong capabilities in semantic understanding and linguistic analysis. The dataset, consisting of news headlines and corresponding sentences, was processed by sending structured prompts to the Claude API, which returned binary labels—0 for relevant and unambiguous sentences, and 1 for potentially ambiguous or contextually irrelevant ones. This approach was chosen for two main reasons: first, Claude 3.5 Sonnet exhibits advanced reasoning and contextual analysis capabilities; and second, the use of LLMs enables large-scale annotation that is both consistent and efficient, minimizing the subjectivity commonly found in manual labeling. The reliability of this method has also been supported by prior research, where Claude 3.5 Sonnet achieved over 90% accuracy in automatic grammaticality analysis on test datasets [29], and demonstrated reliable scoring performance comparable to human raters based on Rasch model analysis [30]. Based on this evidence, the outputs generated by Claude 3.5 are considered valid as ground truth for training and evaluating ambiguous sentence classification in the Indonesian news corpus.

The dataset labeling process is performed to identify and classify sentences that contain ambiguous meaning (ambiguous labels) and sentences that have clear or unambiguous meaning (unambiguous labels). In this case, the analyzed dataset consists of two main categories: ambiguous sentences and unambiguous sentences. After labeling, 22,224 data records were found to fall into the ambiguous category, while the remaining 77,431 data records were classified as unambiguous. The distribution of words in the dataset used for model training and evaluation is displayed in the total data obtained, which can be used to analyze how the model responds to textual ambiguity.

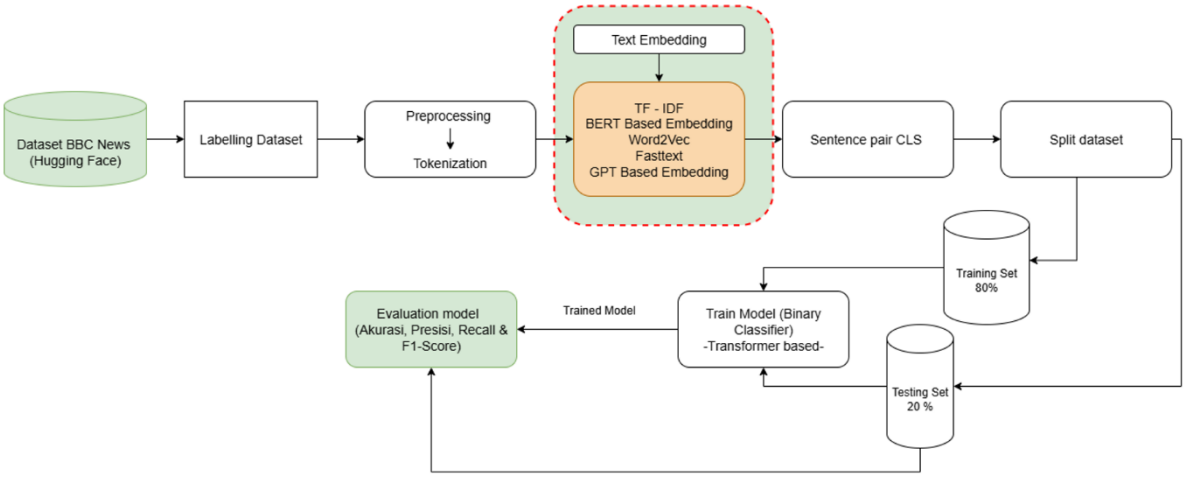

Figure 1. Proposed method

3.3 Preprocessing

In this stage, the preprocessing technique uses tokenization as an important step in text analysis to prepare raw data for processing by the model, breaking the text into sentences, making it easier for the model to understand the structure and context of the text. The result of tokenization is a more structured text. With this structured representation, the model may perform better at identifying ambiguity and categorizing text according to its context [3].

3.4 TF-IDF

In ambiguous and unambiguous classification experiments in text, TF-IDF is used as a method to convert text into a numerical representation that can be used by machine learning algorithms [31]. TF-IDF helps extract important features from text data by weighting words based on their frequency of occurrence in a particular document and their relevance to the overall data set [32].

3.5 BERT based embedding

BERT-based embedding is a transformer-based text representation that understands word context bidirectionally, making it highly effective in the classification of ambiguous and unambiguous text. In this experiment, BERT is used to generate a rich embedding that includes both semantic and syntactic information from the text, allowing the model to capture subtle differences based on word and phrase context. This representation helps machine learning models, such as Random Forest or Logistic Regression and other binary classifier algorithms to distinguish ambiguous text with multi-interpretive words or complex phrases from clearer unambiguous text [33-35].

3.6 Word2Vec

Word2Vec is a neural network-based learning method called word embedding that uses words as numerical vectors in a continuous space. Word2Vec has two primary methods: Skip-gram and Continuous Bag of Words (CBOW). In order to capture the semantic relationship between words, CBOW predicts the target word based on the surrounding context, and Skip-gram predicts the context word based on the target word [36, 37].

3.7 FastText

FastText embedding is a text representation technique that generates a vector for each word, enabling text classification by capturing semantic and morphological relationships between words, including ambiguous text. Compared to other embedding methods, FastText can handle out-of-vocabulary words because it is subword-based, making it superior in capturing patterns in texts with language variations. In its application, the generated word embeddings are averaged to form sentence-level embeddings, which are then used as input features for the classification model in the binary classifier algorithm [38-40].

3.8 GPT-based embedding

Using pre-trained transformer models, such GPT (Generative Pre-trained Transformer), to produce high-level semantic representations of text is known as GPT-based embedding for ambiguous text categorization. GPT models, which are trained on large amounts of natural language data, are capable of capturing complex context, relationships between words, and semantic nuances in text. In classification tasks, the input text is processed by GPT to generate an embedding or vector representation [41]. This embedding is then used as a feature in binary classifier models such as Logistic Regression, Random Forest, Gaussian-NB, etc., to distinguish ambiguous and unambiguous text. The primary benefit of GPT-based embedding is its context awareness, which makes it highly efficient when dealing with ambiguous material. However, because this method uses a lot of computing power, a suitable infrastructure is needed to implement it [42, 43].

3.9 Justification of embedding model selection

This research carefully selects five representative text embedding models TF-IDF, Word2Vec, FastText, BERT, and GPT to comprehensively evaluate various semantic representation approaches in ambiguous sentence classification. TF-IDF offers a simple, computationally efficient, and easy-to-interpret frequency-based representation. Although it is unable to capture semantic context or word order, TF-IDF is important to compare how much added value is provided by more complex embedding models. Word2Vec, as an early-generation neural embedding model, builds a semantic representation based on word co-occurrence patterns. Although efficient and widely used, Word2Vec produces a static representation that does not reflect the variation of word meaning in different contexts. FastText is an extension of Word2Vec that takes into account subword information over n-grams of characters, thus being able to handle languages with complex morphology such as Bahasa Indonesia. BERT enables bidirectional context modeling, where the embedding of a word is influenced by all words in the sentence. This research uses BERT which has been trained on the Indonesian corpus so as to improve the understanding of the unique linguistic structure of Indonesian. Meanwhile, GPT generates contextual embedding through a unidirectional attention mechanism based on autoregressive language modeling. Although different from BERT, GPT embedding is able to build a rich semantic representation by capturing meaning dependencies between words over long ranges. The use of GPT aims to explore the potential of generative models in discriminative classification tasks, particularly for sentence ambiguity detection.

The embedding model selection strategy in this study considers a balance between semantic coverage, computational efficiency, adaptability to available data, and ease of implementation. While we recognize the empirical superiority of models such as ELMo and RoBERTa in various NLP benchmarks [44-46], the focus of this research remains on methods that are in line with resource constraints, interpretation needs, and deployment efficiency. As recommended by a number of recent studies, the use of static embedding and subwords such as TF-IDF, Word2Vec, and FastText is still very relevant and competitive in many NLP tasks, especially in non-generative domains and resource-constrained systems.

3.10 Classification using binary classifier

Classification using a binary classifier is the process of

classifying data into two categories or classes, e.g., Ambiguous and Unambiguous in the context of text ambiguity detection. Some of the algorithms used for binary classification tasks include Logistic Regression, BaggingClassifier, Random Forests, Multinomial Naive Bayes (Multinomial-NB), and Gaussian Naive Bayes (Gaussian-NB).

(1) Logistic Regression (LR)

By creating probabilities for two groups, the regression procedure known as Logistic Regression is employed for binary classification. For ambiguous or unambiguous classification, this model uses a logistic (sigmoid) function to transform the linear output into a value between 0 and 1 [47].

(2) Gaussian Naive Bayes (GNB)

When attributes in a data set are assumed to follow a Gaussian (normal) distribution and have a continuous distribution, the Gaussian Naive Bayes (GNB) form of the Naive Bayes method is employed. In this algorithm, each feature or attribute is viewed as a sample from a normal (Gaussian) distribution that has a certain mean and variance for each class [48].

(3) Multinomial Naive Bayes (MNB)

Multinomial Naive Bayes (MNB) is a probabilistic generative approach used in language modeling, which assumes that linguistic features (e.g., words in a text) are conditionally independent of each other, given a given target class. This approach is simple and highly scalable, allowing it to be used on datasets with a very large number of classes, in contrast to more complex discriminative classifiers [49].

(4) Random Forest (RF)

Several decision trees are combined in the Random Forest ensemble method to provide decisions that are more reliable and robust. Random Forest works well with complicated or unstructured data since each tree is constructed by choosing a random selection of attributes and data. Random Forest is capable of handling more intricate word-context interactions in text classification [50].

(5) Bagging algorithm

An ensemble approach called Bagging (Bootstrap Aggregating) combines several classifiers that have been trained on various training sets in an effort to enhance prediction performance. This approach maintains the same sample size while training each classifier on a training subset that is obtained through redeployment from the original training set. This method creates variance among individual classifiers by using basic random sampling with replacement. The final forecast is then determined by combining the output from each classifier using either weighted majority voting or majority voting. This technique effectively lowers variance and raises the ensemble model's overall accuracy [51].

3.11 Evaluation

In machine learning, the evaluation matrix is used to gauge how well the model performs when completing the classification task. This assessment aids in comprehending the model's ability to forecast the right result and its ability to manage various mistake kinds, as evidenced by accuracy, precision, recall, and F1-Score [32].

$$Precision = \frac{TP}{TP + FP} \quad (1)$$

$$Recall = \frac{TP}{TP + FN} \quad (2)$$

$$F1 - score = 2 * \frac{Precision * Recall}{Precision + Recall} \quad (3)$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (4)$$

The confusion matrix approach is a very effective way to calculate classification performance, such as accuracy, precision, recall, and F1-score. The anticipated outcomes can be used to validate the accuracy values that are used to gauge classification performance based on the confusion matrix [3, 52, 53].

4. RESULT AND ANALYSIS

4.1 Experiment result

4.1.1 Performance of binary classifier model without using text embedding model

This ambiguity classification experiment was conducted to compare several binary classifier algorithms, Specifically, bagging classifier, Random Forest, Logistic Regression, Gaussian Naive Bayes (Gaussian-NB), and Multinomial Naive Bayes (Multinomial-NB). The experiment's primary goal is to assess each algorithm's capacity to identify the BBC News dataset's ambiguous (label 1) and unambiguous (label 0) classes. To give a thorough study of the classification findings, performance metrics are based on precision, recall, f1-score, and confusion matrix. The comparison of the binary classifier algorithms' performance is displayed in Table 1.

Table 1. Performance classification report using binary classifier

Binary Classifier Models	Accuracy	Precision (Unambiguous)	Recall (Unambiguous)	F1-Score (Unambiguous)	Precision (Ambiguous)	Recall (Ambiguous)	F1-Score (Ambiguous)
Random Forest	0.87	0.867	0.987	0.924	0.895	0.415	0.567
Multinomial NB	0.825	0.826	0.987	0.899	0.8	0.195	0.314
Logistic Regression	0.83	0.831	0.99	0.90	0.818	0.22	0.346
Gaussian NB	0.845	0.864	0.956	0.907	0.708	0.415	0.523
Bagging	0.86	0.883	0.95	0.915	0.724	0.512	0.600

Table 2. Performance of the text embedding model on the binary classifier algorithm

Text Embedding	Accuracy	Precision (Unambigous)	Recall (Unambigous)	F1-Score (Unambigous)	Precision (Ambigous)	Recall (Ambigous)	F1-Score (Ambigous)
TF-IDF + Logistic Regression	0.81	0.81	0.99	0.89	0.86	0.14	0.24
TF-IDF + Gaussian Naive Bayes	0.82	0.84	0.96	0.90	0.67	0.33	0.44
TF-IDF + Multinomial Naive Bayes	0.81	0.81	1.00	0.89	1.00	0.10	0.17
TF-IDF + Random Forest	0.81	0.83	0.96	0.89	0.61	0.26	0.37
TF-IDF + Bagging Classifier	0.83	0.84	0.96	0.90	0.70	0.33	0.45
BERT + Logistic Regression	0.81	0.85	0.91	0.88	0.55	0.40	0.47
BERT + Gaussian Naive Bayes	0.73	0.88	0.75	0.81	0.40	0.62	0.49
BERT + Multinomial Naive Bayes	0.79	0.84	0.89	0.87	0.48	0.38	0.43
BERT + Random Forest	0.79	0.80	0.98	0.88	0.50	0.07	0.12
BERT + Bagging Classifier	0.79	0.82	0.94	0.87	0.47	0.21	0.30
Word2Vec + Logistic Regression	0.79	0.79	1.00	0.88	0.00	0.00	0.00
Word2Vec + Gaussian Naive Bayes	0.59	0.90	0.54	0.67	0.31	0.79	0.45
Word2Vec + Multinomial Naive Bayes	0.77	0.79	0.97	0.87	0.17	0.02	0.04
Word2Vec + Random Forest	0.82	0.83	0.97	0.90	0.71	0.24	0.36
Word2Vec + Bagging Classifier	0.81	0.83	0.95	0.89	0.60	0.29	0.39
FastText + Logistic Regression	0.79	0.79	1.00	0.88	0.00	0.00	0.00
FastText + Gaussian Naive Bayes	0.67	0.87	0.69	0.77	0.34	0.60	0.43
FastText + Multinomial Naive Bayes	0.79	0.79	1.00	0.88	0.00	0.00	0.00
FastText + Random Forest	0.81	0.83	0.96	0.89	0.65	0.26	0.37
FastText + Bagging Classifier	0.81	0.84	0.94	0.89	0.61	0.33	0.43
GPT + Logistic Regression	0.82	0.82	0.99	0.90	0.88	0.17	0.28
GPT + Gaussian Naive Bayes	0.80	0.92	0.82	0.87	0.52	0.71	0.60
GPT + Multinomial Naive Bayes	0.74	0.88	0.78	0.83	0.42	0.60	0.50
GPT + Random Forest	0.83	0.84	0.97	0.90	0.75	0.29	0.41
GPT + Bagging Classifier	0.82	0.82	0.99	0.90	0.88	0.17	0.28

Table 1 shows the performance comparison of the binary classifier models in identifying two categories the ambiguous class (label 1) and the unambiguous class (label 0) is displayed in Table 1. With the highest accuracy of 87% and the best precision, recall, and F1-Score in both categories, the Random Forest model is the clear winner and exhibits a superb performance balance. This makes Random Forest particularly suitable for ambiguous text classification tasks as it is able to capture complex patterns in the data. Bagging Classifier came in second with 86% accuracy, and although slightly below Random Forest, it also showed balanced performance in both categories, making it a competitive alternative. In contrast, simple probabilistic models such as Gaussian Naive Bayes and Multinomial Naive Bayes had lower accuracies of 84.5% and 82.5%, respectively, and both experienced significant difficulties in detecting ambiguous text.

4.1.2 Performance of binary classifier algorithm after integration of text embedding model for ambiguous text classification

At this stage, the performance of the binary classification algorithm is evaluated after integrating the text embedding model into the ambiguous text classification process. The main objective, as shown in Table 2, is to analyze whether the embedding technique can reduce classification errors in ambiguous texts.

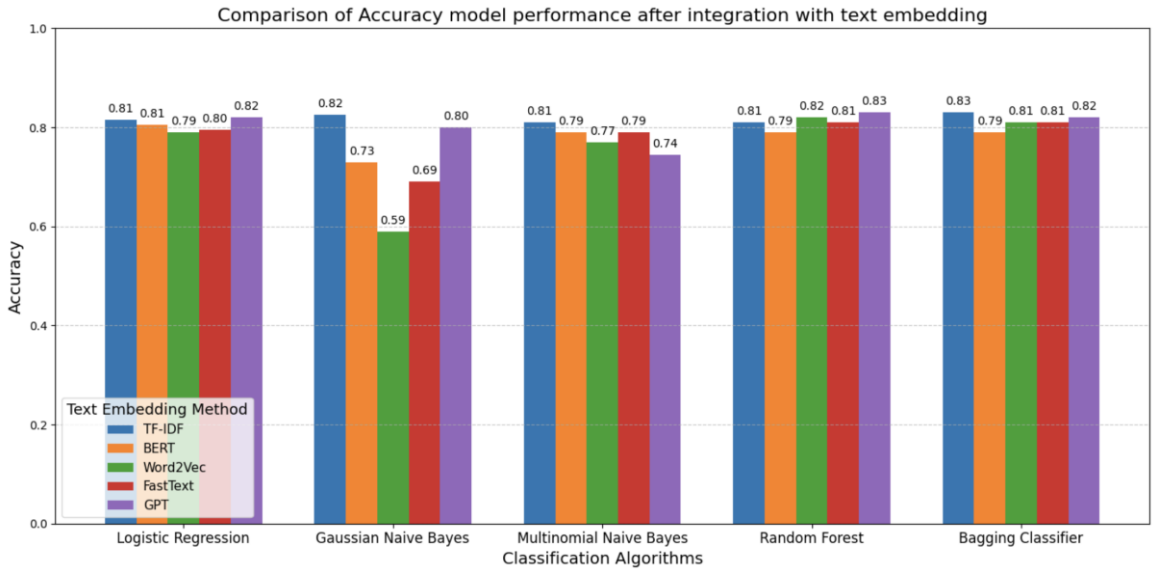

Figure 2. Performance of text embedding model on using binary classifier algorithm

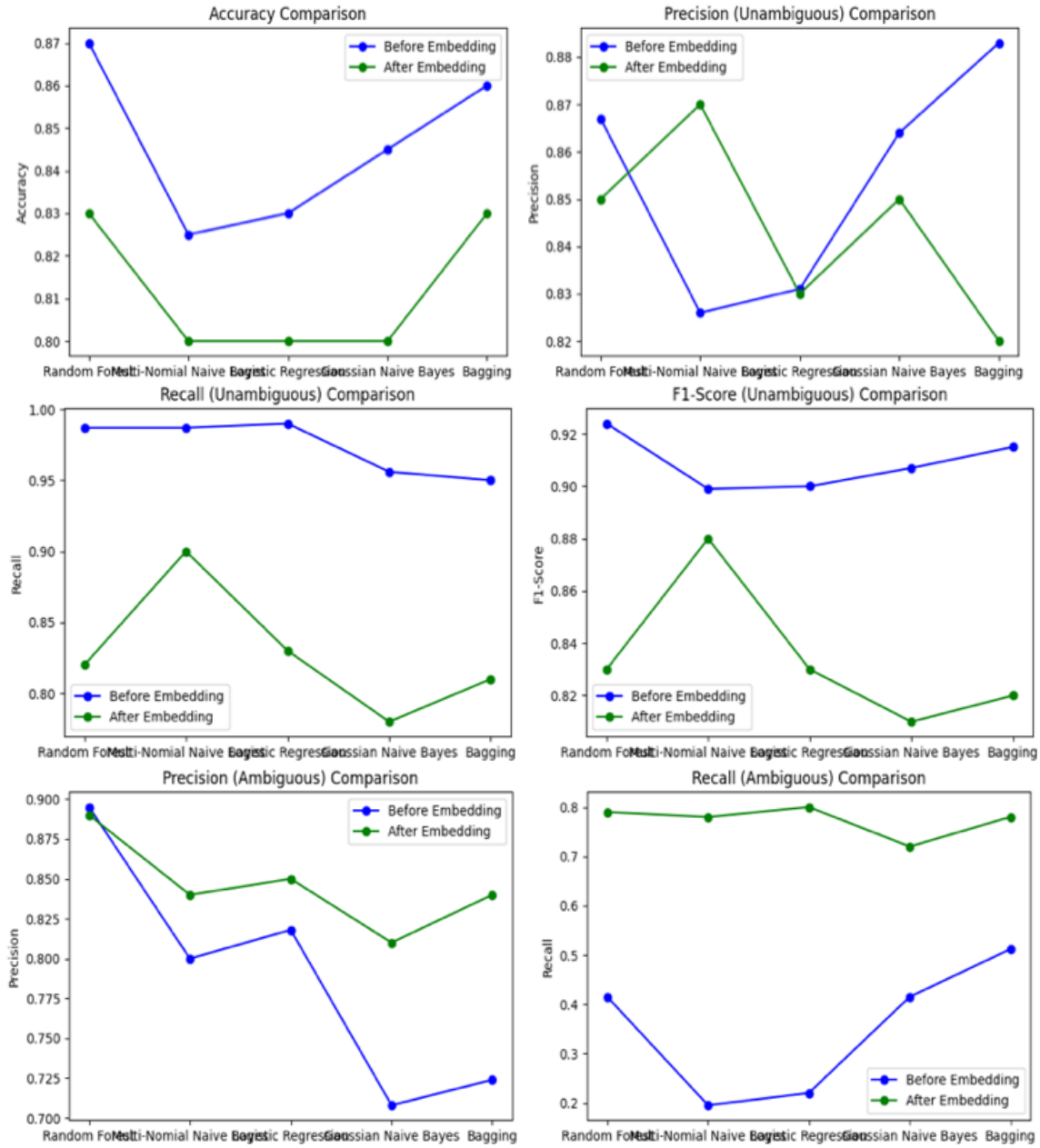

Figure 3. Performance of binary classifier before and after integration of embedding technique

Table 2 shows that the integration of text embedding improves classification performance, especially on ambiguous text. Before embedding, simple representations such as TF-IDF yielded high performance on unambiguous text but struggled with ambiguous text, with low precision and recall. After using embeddings such as GPT and BERT, there was a significant improvement in performance, especially in the GPT + Gaussian Naive Bayes combination, which recorded the highest F1-Score (0.60) for ambiguous text. Embedding GPT and BERT proved to be more effective compared to Word2Vec and FastText, which still showed weaknesses on ambiguous text despite performing well on unambiguous text. Algorithms such as Gaussian Naive Bayes excel at capturing ambiguous context, while methods such as bagging and Random Forests tend to be stronger on unambiguous text. Overall, GPT is the best embedding with the most balanced performance, suggesting that richer text representations can improve classification, especially for ambiguous cases.

Based on Figure 2, the integration of various text embedding methods with different classification algorithms results in competitive accuracy performance. Overall, TF-IDF and GPT-based embedding showed the most consistent and high-performing results in almost all algorithms. The highest accuracy was achieved by GPT on Random Forest (0.83) and Bagging Classifier (0.83), and by TF-IDF on Gaussian Naive Bayes (0.82) and Bagging Classifier (0.82). Meanwhile, BERT maintained a stable performance in the range of 0.77-0.81 but did not consistently outperform the others. In contrast, FastText showed the lowest performance, especially with Gaussian Naive Bayes (0.59), and tended to underperform the other embedding methods in most classifications. Word2Vec also shows relatively lower accuracy, especially on Gaussian Naive Bayes (0.69), although it remains competitive in some cases. The integration of GPT or TF-IDF embedding with binary classifier algorithms such as Random Forest and Bagging Classifier resulted in the most optimal classification performance, highlighting that modern and classical embedding methods can be equally effective when paired with the right algorithms.

Figure 3 shows that the integration of embedding techniques in ambiguous text classification provides a significant change in the performance of binary classifier models, especially in detecting ambiguous text that was previously difficult to handle with traditional approaches. Before the use of embedding, algorithms such as Random Forest and Bagging Classifier showed high accuracy (up to 0.87), but their performance tended to be one-sided. The ambiguous precision of Random Forest was quite high (0.895), but the ambiguous recall was very low (0.415), reflecting the limitations of traditional models in capturing the complex context of ambiguous text. Understanding the semantic subtleties of the text significantly improved after embedding, particularly in GPT + Gaussian NB, which obtained the highest ambiguous recall of 0.71 and a more balanced ambiguous F1-Score of 0.60. Embedding-based models such as GPT and BERT successfully captured deeper semantic relationships, making them ideal for context-dependent ambiguity detection tasks. However, the size of the dataset employed can have an impact on the performance of these embedding-based models, which demand a lot of processing power. On the other hand, word frequency-based embedding, such as TF-IDF, remains relevant, especially in combinations such as TF-IDF + Bagging Classifier, which achieves the highest accuracy of 0.83 with a good balance of performance in both ambiguous and unambiguous classes. In contrast, local context-based embeddings such as Word2Vec and FastText show limitations in capturing global semantic relationships, thus are less optimal on small datasets and result in lower ambiguous F1 scores. In conclusion, modern embeddings such as GPT excel in capturing ambiguous context in depth, while simple approaches such as TF-IDF remain relevant due to their efficiency and simplicity, both in terms of performance and computational resources.

4.2 Comparison confusion matrix before and after integrated text embedding

The classification model's prediction outcomes are assessed using a confusion matrix when there are two classes: ambiguous (1) and unambiguous (0). The confusion matrix in this experiment gives a better idea of how the model divides the input into the two groups. Tables 3 and 4 demonstrate how the evaluation matrix was different before and after the embedding approach was added to the binary classifier.

Table 3. Comparison of confusion matrix for each tested binary classifier model

Binary Classifier Models	True Positive (TP)	False Positive (FP)	False Negative (FN)	True Negative (TN)
Random Forest	151	8	20	21
Multinomial NB	152	7	24	17
Logistic Regression	157	2	32	9
Gaussian NB	157	2	33	8
Bagging	157	2	24	17

Table 4. Comparison of confusion matrix after integration of embedding technique for each binary classifier model

Model Binary Classifier	True Positive (TP)	False Positive (FP)	False Negative (FN)	True Negative (TN)
Random Forest	151	7	31	11
Multinomial NB	158	0	38	4
Logistic Regression	157	1	36	6
Gaussian NB	151	7	28	14
Bagging	152	6	28	14

Figure 4 show the comparison of confusion matrix findings before and after embedding reveals that embedding has a mixed effect on the performance of binary classifier models, as seen in Tables 2-4. Gaussian Naive Bayes benefits significantly from embedding, with improved detection of ambiguous text (decreased FN to 28) and unambiguous text (increased TN to 14), indicating a better balance of performance. Multinomial NB also managed to reduce FP to zero, indicating an excellent ability to detect unambiguous text, but suffered a drastic drop in detection of ambiguous text (FN increased to 38). The Random Forest and bagging classifiers, on the other hand, performed worse after embedding; FN significantly increased and TN significantly decreased, suggesting that embedding is less effective for this model. Logistic Regression showed little change, with a decrease in

performance in detecting ambiguous and unambiguous text. Overall, embedding provides significant benefits to models such as Gaussian Naive Bayes but is not necessarily beneficial for all models, especially Random Forest and Bagging Classifier, which show a decrease in performance. Models that make good use of embedding can be used for tasks that require a balance between ambiguous and unambiguous text.

4.3 Analysis

A more thorough understanding of each model's capabilities is provided by the experimental findings, which compare the models' performance in identifying unambiguous and ambiguous texts before and after the embedding technique was incorporated into the binary classifier algorithm. Table 4 shows the performance comparison on each binary classifier algorithm.

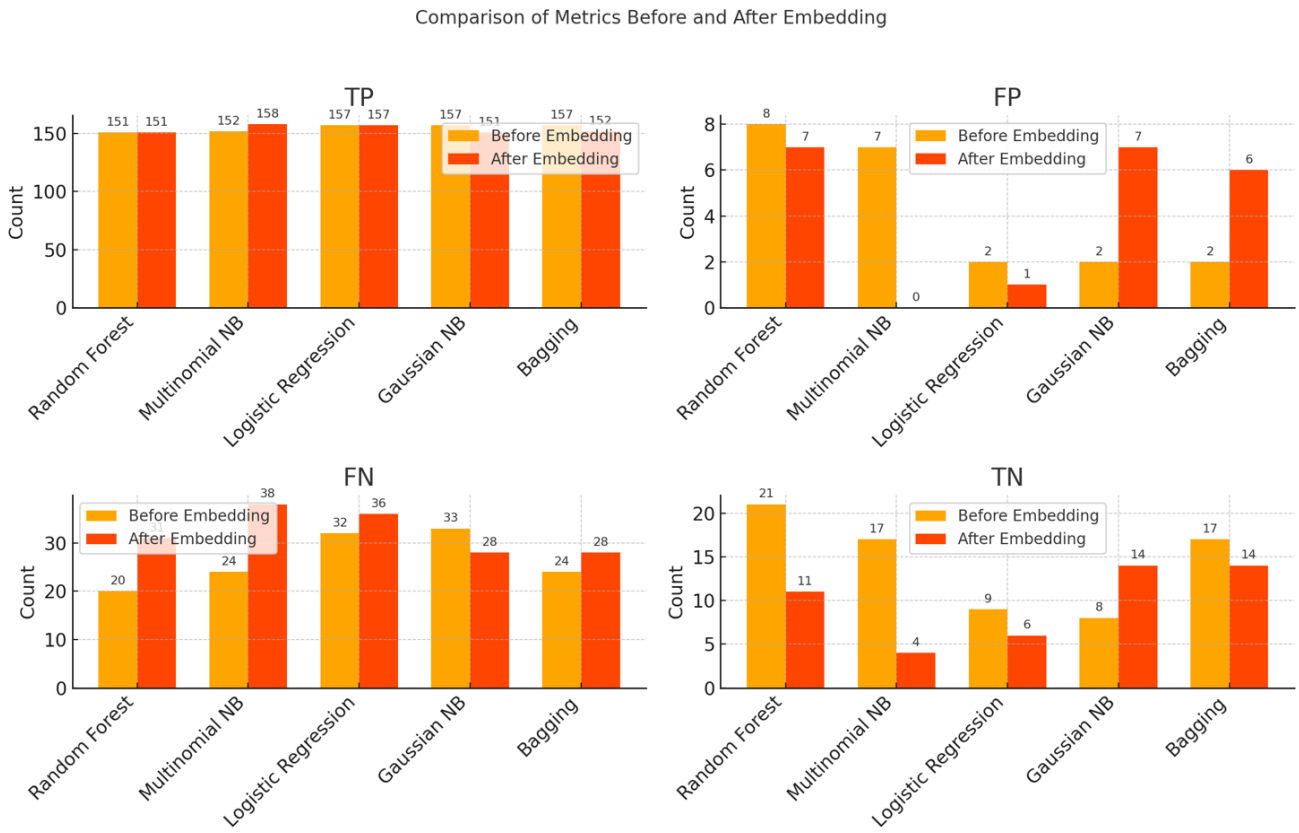

Figure 4. Comparison of confusion matrix before and after integration with embedding technique in binary classifier

Table 5. Model performance analysis

Aspects	Without Embedding	With Embedding
Total Accuracy	Accuracy Up to 0.87 (Random Forest), showing the best performance for overall accuracy without utilizing complex text representations.	Up to 0.83 (TF-IDF + Bagging), it remains competitive despite using a simple word frequency-based approach.
F1-Score Ambigous	Maximum 0.60 (Bagging Classifier), showing the best balance between precision and ambiguous recall before embedding is used.	Maximum 0.60 (GPT + Gaussian NB), equivalent to Bagging but with significantly improved recall for ambiguous text.
Recall Ambigous	Up to 0.415 (Bagging Classifier), indicating the traditional model struggles to detect ambiguous text consistently.	Up to 0.71 (GPT + Gaussian NB), reflecting the ability of modern embedding to capture deeper semantic context.
Precision Ambigous	High in some models, such as Random Forest (0.895), but not matched by sufficient ambiguous recall.	More balanced, especially in GPT + Gaussian NB, which remains competitive despite the increase in recall.
Computational Complexity	Low, utilizing traditional algorithms that are fast and efficient for small datasets or simple tasks.	High, requiring large computational resources for embedding such as GPT and BERT, especially on large datasets.
Data Dependency	Low, can work with small datasets without significant performance degradation.	High, embedding-based models such as GPT and BERT require large datasets to produce optimal text representations.
Usage Efficiency	It is suitable for simple tasks, especially if limited computing resources are a major consideration.	It is suitable for simple tasks, especially if limited computing resources are a major consideration.

Based on Table 5, TF-IDF embedding works well for unambiguous text, delivering high accuracy in models like Bagging and Random Forest, but struggles with ambiguous text, showing low recall and F1-scores. Transformer-based embeddings, such as GPT and BERT, perform better for ambiguous text detection, with GPT + Gaussian Naive Bayes achieving a recall of 0.71 and a balanced F1-score. However, these embeddings are resource-intensive. Random Forest models may not perform well with transformer embeddings due to their complexity. Local embeddings like Word2Vec and FastText capture limited context but fail to grasp global semantics, leading to high FN for ambiguous text. In conclusion, the right combination of embedding and model is essential for optimizing performance, especially in detecting ambiguity.

This study confirms that the integration of modern contextual embedding such as BERT and GPT significantly improves the ability to detect ambiguous sentences, outperforming simple statistical representation-based techniques such as TF-IDF as well as static embedding models such as Word2Vec and FastText. The performance of GPT + Gaussian NB managed to obtain recall up to 0.71 with F1-score of 0.60 for ambiguous classes, while the combination of TF-IDF + Bagging showed the best accuracy for unambiguous classes. Similar conclusions are found in many studies across various language domains, where BERT has been shown to consistently overtake TF-IDF and classical word embeddings in a variety of classification tasks, including hate speech detection, and sentiment analysis. The optimal performance of contextual embeddings occurs due to its ability to represent semantic dependencies between words dynamically based on the sentence context, not just based on co-occurrence patterns or word frequencies like the classical model.

The performance evaluation of text embedding models, particularly BERT and GPT, for ambiguity classification within news corpora has garnered significant research interest, as evidenced by recent comparative analyses. Provide a comprehensive comparison of large language models (LLMs), including BERT and GPT-4, emphasizing their representational capabilities in classifying medical discussions on social media. Their findings highlight the importance of selecting appropriate models based on the specific task, such as ambiguity detection, and demonstrate that transformer-based models can effectively capture nuanced textual features [54]. contribute to this discourse by utilizing a fine-tuned BERT-based model, BERTimbau, for classifying news articles in Portuguese. Their approach involved restructuring categories to mitigate class imbalance, which is a common challenge in ambiguity classification tasks. The success of BERT in this context underscores its robustness in generating meaningful embeddings for complex news texts, facilitating more accurate classification outcomes [55]. offer a nuanced perspective by questioning the assumption that larger pre-trained models like transformers are always optimal for text classification tasks. Their results suggest that the choice of classifier should be task-specific, implying that models like BERT and GPT need to be evaluated carefully within the context of ambiguity detection in news corpora. This aligns with the broader understanding that model performance is contingent upon the nature of the dataset and the classification challenge [56]. explore transliteration-based models for Tamil text summarization, illustrating the broader applicability of advanced embedding techniques in multilingual contexts, which can inform approaches for English news corpora [57].

4.4 Limitation and future works

This research significantly advances the classification of ambiguous sentences in the Indonesian news corpus. However, several limitations exist. First, transformer-based models like BERT and GPT demand high computational resources, posing challenges for large-scale implementation in low-resource environments. Second, automatic labeling using Claude 3.5 Sonnet, despite its strong semantic capabilities, may introduce bias due to prompt sensitivity. Nonetheless, it enables efficient and consistent large-scale annotation. Third, the dataset is limited to news texts, reducing the generalizability of results. Future studies should include human-based validation and explore diverse text domains for broader applicability.

5. CONCLUSION

The results show that the selection of algorithms and embedding techniques has a significant influence on the performance of the model in classifying ambiguous and unambiguous sentences. The best overall results were obtained from ensemble algorithms, specifically Random Forest and Bagging Classifier, with accuracies of 87% and 86%, respectively. Both models showed high precision and recall in classifying unambiguous text. However, their performance degrades when handling ambiguous text, where Random Forest's precision drops to 58%, while Bagging shows better results with 72%. Traditional embedding techniques such as TF-IDF proved to be effective for unambiguous text but less than optimal in handling ambiguous text. In comparison, newer embedding methods like GPT and BERT, along with word-based embeddings like Word2Vec and FastText, offer a deeper understanding of meaning and are better at identifying unclear text. The combination of GPT embedding with Gaussian Naive Bayes yields promising performance on ambiguous text, with a recall of 71% and a balanced F1-score of 60%. BERT embedding was also able to capture context in depth, although the results were slightly less consistent than GPT. Meanwhile, Word2Vec and FastText showed limitations in capturing global semantic relationships, leading to fluctuating performance on ambiguous categories. Binary classification models like Random Forest and Bagging are well-suited for unambiguous text when paired with simple embeddings like TF-IDF. For highly ambiguous text, transformer-based embeddings like GPT and BERT offer better performance, particularly with probabilistic models like Gaussian Naive Bayes.

REFERENCES

[1] Ogaib, M.F., Hashim, K.M. (2022). News classification by N-gram and machine learning algorithms. Journal of Education for Pure Science, 12(2): 327-333. https://doi.org/10.32792/jeps.v12i2.202

[2] Ahmed, J., Ahmed, M. (2021). Online news classification using machine learning techniques. IIUM Engineering Journal, 22(2): 210-225. https://doi.org/10.31436/iiumej.v22i2.1662

[3] Sutriawan, S., Andono, P.N., Muljono, M., Pramunendar, R.A. (2023). Performance evaluation of classification algorithm for movie review sentiment analysis. International Journal of Computing, 22(1): 7-14.

https://doi.org/10.47839/ijc.22.1.2873
[4] Ritzkal, Sutriawan, Prakoso, B.A., Fanani, A.Z., Riawan, I., Fajri, H., Basuki, R.S., Alzami, F. (2023). Word search with trending reviews on Twitter. Ingénierie des Systèmes d'Information, 28(2): 351-356. https://doi.org/10.18280/isi.280210
[5] Sutriawan, S., Rustad, S., Shidik, G.F., Pujiono, P., Muljono, M. (2024). Review of ambiguity problem in text summarization using hybrid ACA and SLR. Intelligent Systems with Applications, 22: 200360. https://doi.org/10.1016/j.iswa.2024.200360
[6] Mengle, S.S., Goharian, N. (2008). Using ambiguity measure feature selection algorithm for support vector machine classifier. In Proceedings of the 2008 ACM Symposium on Applied Computing, pp. 916-920. https://doi.org/10.1145/1363686.1363896
[7] Mahajan, R., Kokane, C., Pathak, K., Kodmelwar, M., Wagh, K., Bhandari, M. (2024). Effect of supervised sense disambiguation model using machine learning technique and word embedding in word sense disambiguation. Journal of Electrical Systems, 20(1s): 436-443. https://doi.org/10.52783/jes.783
[8] Osman, M.H., Zahrin, M.F. (2021). AmbiDetect: An ambiguous software requirements specification detection tool. Turkish Journal of Computer and Mathematics Education, 12(3): 2023-2028.
[9] Maurya, A.S., Gupta, B.K. (2023). A comparative analysis of text classification algorithms for pos ambiguity a comparative analysis of text classification algorithms for pos ambiguity using Weka. European Chemical Bulletin, 12(6): 406-416.
[10] Singh, S., Saikia, L.P., Baruah, S. (2021). A study on quality assessment of requirement engineering document using text classification technique. In 2021 Second International Conference on Electronics and Sustainable Communication Systems (ICESC), Coimbatore, India, pp. 1541-1548. https://doi.org/10.1109/ICESC51422.2021.9532736
[11] Hussain, H.M. (2007). Using text classification to automate ambiguity detection in SRS documents. Doctoral dissertation, Concordia University.
[12] Lee, J., Wang, Z., Johnson, A. (2018). Embracing semantic ambiguity to enhance interpretability of complex unstructured machine learning problems. Proceedings of the Association for Information Science and Technology, 55(1): 849-851. https://doi.org/10.1002/pra2.2018.14505501144
[13] Lee, Y.Y., Ke, H., Yen, T.Y., Huang, H.H., Chen, H.H. (2020). Combining and learning word embedding with WordNet for semantic relatedness and similarity measurement. Journal of the Association for Information Science and Technology, 71(6): 657-670. https://doi.org/10.1002/asi.24289
[14] Yi, K. (2011). An empirical study on the automatic resolution of semantic ambiguity in social tags. Proceedings of the American Society for Information Science and Technology, 48(1): 1-10. https://doi.org/10.1002/meet.2011.14504801175
[15] Yepes, A.J. (2017). Word embeddings and recurrent neural networks based on long-short term memory nodes in supervised biomedical word sense disambiguation. Journal of Biomedical Informatics, 73: 137-147. https://doi.org/10.1016/j.jbi.2017.08.001
[16] Mishra, S., Sharma, A. (2019). On the use of word embeddings for identifying domain specific ambiguities in requirements. In 2019 IEEE 27th International Requirements Engineering Conference Workshops (REW), Jeju, Korea (South), pp. 234-240. https://doi.org/10.1109/REW.2019.00048
[17] Shekhar, S., Sharma, D.K., Beg, M.S. (2019). Embedding framework for identifying ambiguous words in code-mixed social media text. In 2019 International Conference on Contemporary Computing and Informatics (IC3I), Singapore, pp. 59-63. https://doi.org/10.1109/IC3I46837.2019.9055679
[18] Thurnbauer, M., Reisinger, J., Goller, C., Fischer, A. (2023). Towards resolving word ambiguity with word embeddings. arXiv preprint arXiv:2307.13417. https://doi.org/10.48550/arXiv.2307.13417
[19] Helaskar, M.N., Sonawane, S.S. (2019). Text classification using word embeddings. In 2019 5th International Conference on Computing, Communication, Control and Automation (ICCUBEA), Pune, India, pp. 1-4. https://doi.org/10.1109/ICCUBEA47591.2019.9129565
[20] Kamkarhaghighi, M., Makrehchi, M. (2017). Content tree word embedding for document representation. Expert Systems with Applications, 90: 241-249. https://doi.org/10.1016/j.eswa.2017.08.021
[21] Jaber, A., Martínez, P. (2021). Disambiguating clinical abbreviations using pre-trained word embeddings. In Healthinf, pp. 501-508. https://doi.org/10.5220/0010256105010508
[22] Charbonnier, J., Wartena, C. (2018). Using word embeddings for unsupervised acronym disambiguation. In Proceedings of the 27th International Conference on Computational Linguistics, pp. 2610-2619.
[23] Loureiro, D., Rezaee, K., Pilehvar, M.T., Camacho-Collados, J. (2021). Analysis and evaluation of language models for word sense disambiguation. Computational Linguistics, 47(2): 387-443. https://doi.org/10.1162/coli_a_00405
[24] Amur, Z.H., Kwang Hooi, Y., Bhanbhro, H., Dahri, K., Soomro, G.M. (2023). Short-text semantic similarity (STSS): Techniques, challenges and future perspectives. Applied Sciences, 13(6): 3911. https://doi.org/10.3390/app13063911
[25] Mahendra, R., Septiantri, H., Wibowo, H.A., Manurung, R., Adriani, M. (2018). Cross-lingual and supervised learning approach for Indonesian word sense disambiguation task. In Proceedings of the 9th Global Wordnet Conference, pp. 245-250.
[26] Faisal, E., Nurifan, F., Sarno, R. (2018). Word sense disambiguation in Bahasa Indonesia using SVM. In 2018 International Seminar on Application for Technology of Information and Communication, Semarang, Indonesia, pp. 239-243. https://doi.org/10.1109/ISEMANTIC.2018.8549824
[27] Lelywiary, C.J.S., Widowati, S., Lhaksamana, K.M. (2019). Deteksi pola ambiguitas struktural pada spesifikasi kebutuhan perangkat lunak menggunakan pemrosesan bahasa alami. Indonesian Journal on Computing (Indo-JC), 4(3): 51-64. https://doi.org/10.34818/INDOJC.2019.4.3.355
[28] Hasan, T., Bhattacharjee, A., Islam, M.S., Samin, K., Li, Y.F., Kang, Y.B., Rahman, M.S., Shahriyar, R. (2021). XL-sum: Large-scale multilingual abstractive summarization for 44 languages. In Annual Meeting of

the Association of Computational Linguistics and International Joint Conference on Natural Language Processing 2021, pp. 4693-4703. https://doi.org/10.18653/v1/2021.findings-acl.413
[29] Morin, C., Marttinen Larsson, M. (2025). Large corpora and large language models: A replicable method for automating grammatical annotation. Linguistics Vanguard. https://doi.org/10.1515/lingvan-2024-0228
[30] Jiao, H., Song, D., Lee, W.C. (2025). Comparing human and AI rater effects using the many-facet Rasch model. https://doi.org/10.48550/arxiv.2505.18486
[31] Kim, D., Seo, D., Cho, S., Kang, P. (2019). Multi-co-training for document classification using various document representations: TF–IDF, LDA, and Doc2Vec. Information Sciences, 477: 15-29. https://doi.org/10.1016/j.ins.2018.10.006
[32] Alamin, Z., Lorosae, T.A., Ramadhan, S. (2024). Improving performance sentiment movie review classification using hybrid feature TFIDF, N-gram, information gain and support vector machine. Mathematical Modelling of Engineering Problems, 11(2): 375-384. https://doi.org/10.18280/mmep.110209
[33] Reimers, N., Gurevych, I. (2019). Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pp. 3982-3992. https://doi.org/10.18653/v1/d19-1410
[34] Wiedemann, G., Remus, S., Chawla, A., Biemann, C. (2019). Does BERT make any sense? Interpretable word sense disambiguation with contextualized embeddings. arXiv preprint arXiv: 1909.10430. https://doi.org/10.48550/arXiv.1909.10430
[35] Wang, Q., Liu, P., Zhu, Z., Yin, H., Zhang, Q., Zhang, L. (2019). A text abstraction summary model based on BERT word embedding and reinforcement learning. Applied Sciences, 9(21): 4701. https://doi.org/10.3390/app9214701
[36] Lin, S., Zhang, R., Yu, Z., Zhang, N. (2020). Sentiment analysis of movie reviews based on improved Word2Vec and ensemble learning. Journal of Physics: Conference Series, 1693(1): 012088. https://doi.org/10.1088/1742-6596/1693/1/012088
[37] Yilmaz, S., Toklu, S. (2020). A deep learning analysis on question classification task using Word2vec representations. Neural Computing and Applications, 32(7): 2909-2928. https://doi.org/10.1007/s00521-020-04725-w
[38] Yao, T., Zhai, Z., Gao, B. (2020). Text classification model based on fastText. In 2020 IEEE International Conference on Artificial Intelligence and Information Systems (ICAIIS), Dalian, China, pp. 154-157. https://doi.org/10.1109/ICAIIS49377.2020.9194939
[39] Mojumder, P., Hasan, M., Hossain, M.F., Hasan, K.M.A. (2020). A Study of fastText word embedding effects in document classification in Bangla Language. In: Bhuiyan, T., Rahman, M.M., Ali, M.A. (eds) Cyber Security and Computer Science. ICONCS 2020. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol 325. Springer, Cham. https://doi.org/10.1007/978-3-030-52856-0_35
[40] Umer, M., Imtiaz, Z., Ahmad, M., Nappi, M., Medaglia, C., Choi, G.S., Mehmood, A. (2023). Impact of convolutional neural network and fastText embedding on text classification. Multimedia Tools and Applications, 82(4): 5569-5585. https://doi.org/10.1007/s11042-022-13459-x
[41] Lee, M. (2024). Fractal analysis of GPT-2 token embedding spaces: Stability and evolution of correlation dimension. Fractal and Fractional, 8(10): 603. https://doi.org/10.3390/fractalfract8100603
[42] Al-Hasan, T.M., Sayed, A.N., Bensaali, F., Himeur, Y., Varlamis, I., Dimitrakopoulos, G. (2024). From traditional recommender systems to GPT-based chatbots: A survey of recent developments and future directions. Big Data and Cognitive Computing, 8(4): 36. https://doi.org/10.3390/bdcc8040036
[43] Lv, H., Niu, Z., Han, W., Li, X. (2024). Can GPT embeddings enhance visual exploration of literature datasets? A case study on isostatic pressing research. Journal of Visualization, 27(6): 1213-1226. https://doi.org/10.1007/s12650-024-01010-z
[44] Tan, K.L., Lee, C.P., Lim, K.M. (2023). Roberta-GRU: A hybrid deep learning model for enhanced sentiment analysis. Applied Sciences, 13(6): 3915. https://doi.org/10.3390/app13063915
[45] Neuraz, A., Rance, B., Garcelon, N., Llanos, L.C., Burgun, A., Rosset, S. (2020). The impact of specialized corpora for word embeddings in natural language understanding. Studies in Health Technology and Informatics, 270: 432-436. https://doi.org/10.3233/SHTI200197
[46] Cai, F., Hu, Q., Zhou, R., Xiong, N. (2023). REEGAT: RoBERTa entity embedding and graph attention networks enhanced sentence representation for relation extraction. Electronics, 12(11): 2429. https://doi.org/10.3390/electronics12112429
[47] Jiang, Z., Liu, H., Fu, B., Wu, Z. (2017). Generalized ambiguity decompositions for classification with applications in active learning and unsupervised ensemble pruning. Proceedings of the AAAI Conference on Artificial Intelligence, 31(1): 2073-2079. https://doi.org/10.1609/aaai.v31i1.10834
[48] Agarwal, S., Jha, B., Kumar, T., Kumar, M., Ranjan, P. (2019). Hybrid of Naive Bayes and Gaussian Naive Bayes for classification: A map reduce approach. International Journal of Innovative Technology and Exploring Engineering (IJITEE), 8(6): 266-268.
[49] Chirawichitchai, N. (2013). Sentiment classification by a hybrid method of greedy search and multinomial Naïve Bayes algorithm. In 2013 Eleventh International Conference on ICT and Knowledge Engineering, Bangkok, Thailand, pp. 1-4. https://doi.org/10.1109/ICTKE.2013.6756285
[50] Fauzi, M.A. (2018). Random forest approach for sentiment analysis in Indonesian. Indonesian Journal of Electrical Engineering and Computer Science, 12(1): 46-50. https://doi.org/10.11591/ijeecs.v12.i1.pp46-50
[51] Onan, A., Korukoğlu, S., Bulut, H. (2016). Ensemble of keyword extraction methods and classifiers in text classification. Expert Systems with Applications, 57: 232-247. https://doi.org/10.1016/j.eswa.2016.03.045
[52] Heydarian, M., Doyle, T.E., Samavi, R. (2022). MLCM: Multi-label confusion matrix. IEEE Access, 10: 19083-19095. https://doi.org/10.1109/ACCESS.2022.3151048
[53] Haryanto, A.W., Mawardi, E.K. (2018). Influence of

word normalization and chi-squared feature selection on support vector machine (SVM) text classification. In 2018 International Seminar on Application for Technology of Information and Communication, Semarang, Indonesia, pp. 229-233. https://doi.org/10.1109/ISEMANTIC.2018.8549748

[54] De Santis, E., Martino, A., Ronci, F., Rizzi, A. (2024). From bag-of-words to transformers: A comparative study for text classification in healthcare discussions in social media. IEEE Transactions on Emerging Topics in Computational Intelligence, 9(1): 1063-1077. https://doi.org/10.1109/TETCI.2024.3423444

[55] Santana, I.N., Oliveira, R.S., Nascimento, E.G. (2022). Text classification of news using transformer-based models for Portuguese. Journal of Systemics, Cybernetics and Informatics, 20(5): 33-59. https://doi.org/10.54808/JSCI.20.05.33

[56] Siino, M., Di Nuovo, E., Tinnirello, I., La Cascia, M. (2022). Fake news spreaders detection: Sometimes attention is not all you need. Information, 13(9): 426. https://doi.org/10.3390/info13090426

[57] Dhivyaa, C.R., Nithya, K., Janani, T., Kumar, K.S., Prashanth, N. (2022). Transliteration based generative pre-trained transformer 2 model for Tamil text summarization. In 2022 International Conference on Computer Communication and Informatics (ICCCI), Coimbatore, India, pp. 1-6. https://doi.org/10.1109/ICCCI54379.2022.9740991