Proceedings of the 8th International Conference on Computing and Data Science
DOI: 10.54254/2755-2721/2026.36442

Research on Text Information Extraction and Imbalanced Classification Methods for Enterprise Profiling

Xinyi Xu

School of Transportation, Tongji University, Shanghai, China
626949947@qq.com

Abstract. This research focuses on enterprise profiling in scenarios where large volumes of diverse texts—such as registration records, annual reports, news articles, and bidding notices—are continuously generated. Instead of relying solely on a single data representation or classification model, we developed a comprehensive natural language processing (NLP) pipeline for extracting key information and identifying industries. The pipeline consists of several steps. First, we use a BERT-BiLSTM-CRF model to identify core e/nterprise entities. Then, we combine TF-IDF with BERT embeddings to create a hybrid feature scheme that captures both lexical cues and contextual semantics. To address the challenge of imbalanced industry labels, we apply SMOTE in the dense semantic space and pair it with Focal Loss to enhance learning for minority classes. Additionally, we introduce a Stacking strategy to integrate outputs from different models, making predictions more stable. Tests on a self-compiled dataset covering ten national economic sectors and about 50,000 enterprises show that our method achieves a macro-F1 score of 95.4%. It outperforms traditional machine learning baselines and single deep learning models, offering more reliable recognition for minority classes. These results suggest that our framework is well-suited for applications such as supply chain partner discovery, industrial mapping, and targeted investment promotion.

Keywords: Natural language processing, enterprise information extraction, text classification, BERT

1. Introduction

China's digital economy has expanded rapidly in recent years, and reforms in the business-registration system have made enterprise information far more accessible than before. As disclosure mechanisms have improved, large volumes of enterprise-related text have accumulated across registration records, business-scope statements, annual reports, bidding notices, judicial documents, and news coverage. The National Enterprise Credit Information Publicity System alone contains more than 170 million market entities, and both structured and unstructured records continue to increase by millions each day. These materials contain rich signals about firm attributes, operating patterns, and interindustry linkages, which makes them valuable for enterprise profiling, industrial mapping, supply-chain analysis, and targeted investment promotion. Yet enterprise text is difficult to process because it mixes technical terminology, overlapping industry expressions, varied writing

DOI: 10.54254/2755-2721/2026.36442

styles, and strongly imbalanced category distributions. As a result, manual annotation and rule-based matching alone are no longer sufficient for efficient and reliable analysis. Meanwhile, advances in deep learning and pretrained language models have opened new possibilities for large-scale text understanding. Models trained on massive unlabeled corpora can learn contextual semantics and then be adapted to downstream tasks with limited labeled data, which has already improved text classification, named entity recognition, and relation extraction. These developments make NLP a practical route for uncovering hidden industrial relationships and supporting smarter enterprise analysis under current policy priorities for digital transformation.

In both research and practical applications, automatically extracting and classifying enterprise information is crucial. In supply chain management, precisely identifying enterprise types enables core companies to quickly select upstream and downstream partners, enhancing coordination efficiency and supply chain resilience. In industrial economics, such classification facilitates more detailed analysis of industrial clusters, regional economic structures, and policy impacts. This type of information is equally valuable in public administration and financial services. Enterprise profiles can support targeted investment promotion, regulatory oversight, credit evaluation, risk warnings, anti-fraud efforts, and lending decisions. Therefore, improving the extraction and classification of enterprise text information remains of significant academic and practical importance.

Recent research has generated a substantial amount of work in the areas of text classification and information extraction. Singh et al. [1] examined information extraction techniques based on natural language processing (NLP) and highlighted the ongoing shift from manual rule creation and statistical features to deep learning approaches. Later, Devlin et al. [2] demonstrated that using a pretraining-and-fine-tuning approach could significantly enhance performance in classification and named entity recognition tasks. Vaswani et al. [3] introduced the Transformer architecture, which incorporates the self-attention mechanism that is now a key component of many large-scale pretrained models. However, much of this research has primarily focused on general-domain datasets. Enterprise text presents unique challenges that are not fully addressed by existing studies. These challenges include integrating data from diverse sources, dealing with semantic overlaps across different industries, managing severe data imbalances, and adapting models to specific business contexts.

To tackle these challenges, this paper constructs an end-to-end framework for extracting and classifying enterprise text. It evaluates the performance of TF-IDF, Word2Vec [4], and BERT representations when paired with four different classifiers, allowing for an assessment of technical options under various application constraints. The study also integrates SMOTE, Focal Loss, and Stacking techniques to enhance the recognition of minority classes while preserving overall accuracy. Finally, the proposed method is tested in a simulated supply-chain environment, demonstrating how classification results can assist in enterprise positioning and profile development in business contexts.

2. Related work and theoretical foundations

2.1. Natural language processing technologies

Natural language processing (NLP) has undergone several clear developmental stages. Initially, rule-based systems dominated, relying heavily on manually crafted lexicons [5]. These systems often delivered high precision but were challenging to maintain and struggled with generalization. Subsequently, statistical learning methods gained prominence. Techniques like hidden Markov models, conditional random fields, and maximum entropy models became standard for tasks such as

DOI: 10.54254/2755-2721/2026.36442

text segmentation, part-of-speech tagging, and named entity recognition. The field then shifted toward deep representation learning, introducing distributed embeddings such as Word2Vec and GloVe. These methods encoded semantic information in continuous vector spaces, moving away from sparse one-hot representations. Sequence models like LSTM, GRU, and the Transformer further advanced NLP by improving the handling of long-range contextual dependencies. Since the introduction of BERT, pretrained language models have become the primary approach in NLP. More recently, large models such as GPT-4 and LLaMA have significantly enhanced zero-shot and few-shot learning capabilities [6].

2.2. Text classification algorithms

Text classification is a classic task in natural language processing (NLP), and its algorithmic framework can be broadly divided into three categories. The first category consists of traditional machine learning methods, represented by support vector machines [7], naive Bayes, logistic regression, and random forests [8]. These methods are often combined with TF-IDF or n-gram features, performing stably on small to medium-sized datasets while offering strong interpretability. The second category includes shallow deep learning approaches. For instance, Kim's [4] TextCNN model extracts local n-gram semantic features through multi-scale convolutional kernels, achieving good results in tasks like sentiment analysis and news classification. Additionally, the BiLSTM-Attention model, based on bidirectional long short-term memory networks (BiLSTM) and attention mechanisms, has been widely applied. The third category comprises methods leveraging pre-trained language models, where models such as BERT serve as feature extractors or undergo direct fine-tuning. Even with limited labeled data, these approaches maintain high accuracy and have become the mainstream choice in current industrial applications.

2.3. Enterprise information processing

In enterprise operations, information processing is a critical link. It involves the collection, organization, analysis, and application of various types of enterprise data, aiming to help enterprises better understand market dynamics, optimize business processes, and improve decision-making efficiency. Through effective information processing, enterprises can uncover the value behind data and provide strong support for strategic planning and daily operations. There are two main directions for research on enterprise information processing: information extraction and classification based portrait construction. In terms of information extraction, named entity recognition is a fundamental technology used to identify key elements such as enterprise name, legal representative, industry terminology, etc. Relationship extraction is used to construct semantic relationships between enterprises, such as equity, supply, guarantee, etc., and is an important foundation for building industrial knowledge graphs. In terms of classification and portrait construction, researchers not only focus on industry classification, but also pay attention to the automatic generation of multi-dimensional labels such as enterprise size, lifecycle, and innovation capability. In addition, multiple enterprise portrait commercial products have emerged in the industry, such as Tianyancha, Qichacha, Qixinbao, etc. These products integrate multiple sources of data such as business records, judicial records, intellectual property data, and public opinion data to provide comprehensive profiles for corporate entities, supporting investment and financing, investment attraction, credit investigation, and other decision-making. However, in public academic research, there is still relatively little systematic comparison of different feature representations and

DOI: 10.54254/2755-2721/2026.36442

classification algorithms for enterprise industry classification, which is also one of the main motivations for this study.

3. Overall design

3.1. System architecture design

This article proposes a framework for automatically extracting, classifying, and constructing portraits from enterprise texts. This framework consists of five interrelated parts: data collection, data preprocessing, text representation, model construction, and portrait generation. In the data collection stage, we obtain texts related to enterprises from public channels, such as the National Enterprise Credit Information Publicity System, listed company announcements, industry news, and bidding platforms, in order to construct a domain corpus. Subsequently, the original text is cleaned, including word segmentation, removal of stop words, and recognition of named entities, to ensure that the input for subsequent analysis is more consistent. We use methods such as TF-IDF, Word2Vec, and BERT for feature representation, which can capture different levels of vocabulary and semantic information. Using these text representations, we trained support vector machines (SVM), random forests, TextCNN, and fine tuned BERT classifiers. In addition, stacking technology has been introduced to improve prediction performance. The final output combines industry labels and portrait information, which can be applied to tasks such as enterprise retrieval, supply chain analysis, and industry research.

3.2. Technical route

This study is divided into five primary steps: gathering data, preprocessing text, constructing features, training models, and generating enterprise profiles. Once the text data is collected, it undergoes cleaning processes such as removing duplicates, standardizing formats, segmenting Chinese words, and filtering out stop words. The study also identifies important entities, like enterprise names and business scopes. For representing the text, different feature sets are created using TF-IDF, Word2Vec, and BERT. The effectiveness of these methods is evaluated by comparing their performance in downstream classification tasks, as this step is crucial. In model training, a fine-tuned BERT model serves as the main classifier, while SVM, random forest, and TextCNN are used as benchmarks for comparison. To address the issue of class imbalance, the study employs SMOTE oversampling and Focal Loss techniques. It then uses stacking to combine the predictions of multiple models, aiming to enhance accuracy and stability. In the final step, the system generates the predicted industry category and structured profile information for each enterprise.

4. Data preprocessing and feature engineering

4.1. Data sources

The corpus used in this study includes four types of data: first, basic registration information of enterprises, specifically covering fields such as enterprise name, unified social credit code, registered address, registered capital, and business scope; The second is the annual and regular business reports of listed companies, which include descriptions of main business activities, financial summaries, and shareholder information; Thirdly, industry news announcements reflect the recent business dynamics, investment and financing events, and cooperation agreements of the enterprise; The fourth is the public bidding records, which provide strong basis for actual business

DOI: 10.54254/2755-2721/2026.36442

labels. All data was obtained through compliant channels and desensitized. The setting of target classification labels is based on the industry classification and main categories in the National Economic Industry Classification (GB/T 4754-2017).

4.2. Text cleaning and standardization

There are often issues with excessive spaces, HTML tags, garbled characters, and mixed use of traditional and full width characters in the original text. This study uses regular expressions and character set mapping methods for data cleaning and normalization. The specific steps are: first remove the HTML and XML tags, and convert the data to UTF-8 encoding uniformly; Then convert traditional Chinese to simplified Chinese, and convert full width characters to half width characters; Remove consecutive spaces and invisible characters, normalize numbers and English symbols, and finally obtain a standardized Chinese corpus. In addition, this study also constructed a universal stop word list and added industry-specific stop words based on the characteristics of enterprise texts, filtering out high-frequency but low discriminatory words such as "company", "limited", and "shares".

4.3. Chinese word segmentation and part-of-speech tagging

Due to the lack of natural word boundaries in Chinese text, the quality of word segmentation will affect all subsequent feature constructions. This study compared three commonly used segmentation tools for enterprise business scope data: jieba, HanLP, and pkuseg. Jieba word segmentation tool is lightweight and easy to customize, and can recognize industry terms well after loading domain dictionaries, but its ability to discover new words is weak. HanLP combines methods based on perceptrons and neural networks, resulting in better processing performance for unregistered words, but with slightly slower inference speed. Pkuseg supports domain adaptation and performs excellently on professional corpora. Based on the above characteristics, the final system adopts jieba as the main segmentation tool, supplemented with a self built dictionary of about 32000 industry terms, and verifies uncertain segmentation results with reference to HanLP's output. Part of speech tagging adopts Peking University standards to ensure consistency in subsequent syntactic feature extraction.

4.4. Comparison of feature representation methods

Feature representation sets a practical upper limit on classification performance. This paper compares three representative ways of representing features. The first is TF-IDF,which measures how well a term separates documents by combining term frequency with inverse document frequency. Its features are high-dimensional and sparse,easy to interpret,and especially well suited to linear classifiers. The second is Word2Vec [3], which trains 200-dimensional continuous word vectors on the enterprise corpus and obtains sentence vectors through average pooling. It can capture semantic similarity but is weak in representing context and polysemy. The third is BERT [2], which takes the 768-dimensional vector at the [CLS] position after feeding the text into a pretrained model as the sentence vector. This representation integrates bidirectional context and large-scale prior knowledge and has the strongest semantic representation ability, but it also entails substantial inference overhead and a relatively large model size. To balance the explicit discriminative power of sparse features with the semantic generalization ability of dense features, this paper further experiments with a hybrid representation that concatenates PCA-reduced TF-IDF features with

DOI: 10.54254/2755-2721/2026.36442

BERT vectors. Preliminary results show that, on long-tail categories, the hybrid representation improves the F1 score by about 1.4 percentage points compared with using BERT features alone, verifying the complementarity of sparse and dense features in enterprise industry classification.

4.5. Feature dimensionality reduction and selection

In data analysis and machine learning, dealing with high-dimensional data can be challenging due to the curse of dimensionality. To address this, we often need to reduce the number of feature dimensions and select the most relevant ones. This process helps simplify the model, improve computational efficiency, and enhance prediction accuracy. By reducing dimensions, we aim to keep the essential information while eliminating redundant or less important features. Selection, on the other hand, focuses on picking out the features that contribute the most to the model's performance. Together, these techniques enable us to work with more manageable and effective data sets.

5. Enterprise classification model design and implementation

5.1. Comparison of candidate classification models

This study compares four representative classification models in terms of interpretability,training cost,inference efficiency,and pblueictive accuracy. Support vector machines [6] separate classes by maximizing the margin between them and usually perform well with TF-IDF features,although kernel design and large-scale deployment can still be difficult. In this paper,a one-versus-rest strategy with an RBF kernel is used to improve nonlinear discrimination.

Random forests [7] blueuce variance through bootstrap sampling and multi-tree voting. They also tend to handle noisy features well and can produce feature-importance outputs,which helps with interpretation. TextCNN [8] applies convolution to text classification and captures local n-gram patterns through kernels of different sizes,so it suits short texts such as business scopes.

Fine-tuned BERT represents the pretrained language model approach and is optimized end to end for the task objective. These four models were selected because they cover several common routes in text classification: traditional statistical learning,ensemble learning,shallow neural modeling,and large pretrained models. This gives a more practical basis for technical selection in engineering applications.

5.2. Model training pipeline

We trained the model following a standard machine learning workflow. The dataset was split into training, validation, and test sets with a ratio of 7:1:2. We kept the industry category distribution roughly the same across these sets to prevent data leakage. We used cross-entropy as the primary loss function and AdamW as the optimizer. For fine-tuning BERT, we set the initial learning rate to 2 x 10^-5 and applied a linear warm-up strategy. For TextCNN, the initial learning rate was set to 1 x 10^-3, and we included an early stopping mechanism. The setup was simple. During training, we tracked the training loss, validation accuracy, and macro-F1 score for each epoch. Training stopped early if validation metrics did not improve for three consecutive epochs, and we saved the best model weights for final evaluation.

DOI: 10.54254/2755-2721/2026.36442

5.3. Hyperparameter tuning strategy

This paper compares two strategies for searching hyperparameters: grid search and Bayesian optimization. Grid search involves testing all predefined combinations of parameters. It is easy to implement, but its computational cost increases quickly as the search space gets larger. Bayesian optimization, on the other hand, treats parameter tuning as a step-by-step optimization problem. It uses a surrogate model and an acquisition rule to balance exploring new possibilities and exploiting known good solutions. This approach often finds competitive solutions with far fewer attempts. In experiments fine-tuning BERT, the variables adjusted included batch size, maximum sequence length, learning rate, dropout rate, and weight decay. The results show that a configuration nearly as good as the best one found after 96 grid search runs can be achieved with just 24 iterations of Bayesian optimization, significantly reducing training costs.

5.4. Ensemble learning method

To enhance model generalization, this paper proposes a two-stage ensemble method. The first stage employs soft voting, which calculates the average predictive probabilities from multiple runs of the same model with different random seeds. This approach reduces the impact of random initialization on a single model. The second stage uses Stacking, where cross-validation outputs from four base models—SVM, random forest, TextCNN, and fine-tuned BERT—on the training set serve as meta-features. Logistic regression is then used as the meta-learner to make the final classification decision. To prevent data leakage, the training set is divided into five parts. For each part, meta-features are generated from models trained on data outside that part. This ensures that the feature distribution seen by the meta-learner matches that during testing. Stacking fusion effectively leverages the strengths of each base model: SVM performs consistently with sparse features, random forests are resilient to noise, TextCNN captures local n-gram patterns well, and BERT provides strong semantic context. Combining these models improves the macro-F1 score on the validation set by 1.8 percentage points.

5.5. Handling class imbalance

Enterprise industry classification shows a clear long-tail pattern. Major categories such as manufacturing and wholesale or retail trade contribute more than 40% of the samples, whereas tail categories including agriculture, forestry, animal husbandry, fishery, and public administration each account for less than 2%. To mitigate this imbalance, the study adopts two complementary measures. The first is SMOTE oversampling [9], which synthesizes minority samples in the neighborhood of existing observations to reduce bias in the training distribution. Because text itself is discrete, SMOTE is not applied to raw sentences; instead, it is performed in the dense feature space generated by BERT so that the synthetic vectors stay closer to the semantic distribution of real data. The second measure is Focal Loss [10], which down-weights easy samples and directs more attention to difficult minority examples. When these two strategies are used together, the average F1 score of tail categories rises by 6.3 percentage points without sacrificing the accuracy of major classes.

DOI: 10.54254/2755-2721/2026.36442

6. Experimental results and analysis

6.1. Experimental setup

The experiments are conducted on a dataset containing about 50,000 enterprise samples drawn from ten national economic industry categories. Each record includes information such as the enterprise name, business scope, company profile, and major news summaries from the last three years. Industry labels are assigned independently by three annotators and finalized through majority voting, resulting in a Kappa coefficient of 0.86, which indicates good annotation consistency. Because the dataset retains a realistic long-tail distribution, it is well suited for examining whether the proposed strategy can improve performance on minority classes without weakening the results on major categories. Evaluation relies on accuracy, precision, recall, and F1 score, and both macro and micro averages are reported. Macro-level metrics reflect class-level balance more clearly, whereas micro-level metrics describe overall sample-level performance. Confusion matrices are further used to inspect typical error patterns.

6.2. Experimental results

Table 1. Comparison of experimental results for different feature representations and classification model combinations

Feature Representation	Classification Model	Accuracy (%)	Macro-F1 (%)	Micro-F1 (%)
TF-IDF	SVM	82.5	79.4	83.1
TF-IDF	Random Forest	83.7	80.6	84.2
Word2Vec	Random Forest	85.1	82.1	85.6
Word2Vec	TextCNN	88.9	87.6	89.4
BERT[CLS]	Fully Connected Classifier	94.7	93.6	94.9
BERT+TF-IDF	Stacking Ensemble	96.1	95.4	96.3

The results in Table 1 show significant differences between different feature representations and model combinations. TF-IDF combined with SVM performs stably in the main categories, but has lower recall rates in a few categories, resulting in a macro F1 score of 79.4%. When replacing SVM with a random forest based on Word2Vec features, the macro F1 score increased to 82.1%. TextCNN further increases the score to 87.6% by directly learning local patterns from the embedding matrix. In the single model, the BERT after fine adjustment performs best, with the macro F1 score of 93.6% and the micro F1 score of 94.9%. After combining the outputs of multiple models, the macro F1 score of the Stacking ensemble model reached 95.4%, indicating that the base learner captured different types of information. The overall trend is relatively clear: with the enrichment of feature representation and the increase of model capacity, the overall performance has also been improved. Compared to TextCNN, BERT performs better on semantically more complex categories, which may be due to its pre trained contextual knowledge that helps to handle specialized terminology and long-distance dependencies. In terms of efficiency, there are differences in the average inference time among different models: SVM is about 0.6 milliseconds per sample, random forest is 1.2 milliseconds per sample, TextCNN is 3.8 milliseconds per sample, fine tuned BERT is 22 milliseconds per sample, and Stacking ensemble model is 28 milliseconds per sample. If accuracy is the main requirement in practical applications, although BERT and ensemble models may increase some latency, their performance is still worth considering.

DOI: 10.54254/2755-2721/2026.36442

6.3. Ablation analysis

This paper employs ablation experiments to assess the contribution of each component. When the custom industry lexicon is removed, segmentation accuracy drops by approximately 3.2 percentage points, and the downstream macro-F1 score decreases by 2.1 percentage points. This indicates that specialized vocabulary remains important in enterprise text processing. The impact is evident. Without Focal Loss, the average F1 score for tail categories decreases by 4.8 percentage points. Removing SMOTE oversampling further reduces recall for minority classes by 3.5 percentage points. Replacing the hybrid representation with BERT features alone also leads to a 1.4 percentage point drop in macro-F1. These findings collectively suggest that combining sparse lexical cues with dense semantic information yields better results, and that strategies addressing imbalance enhance recognition of tail categories.

7. Conclusion and future work

In summary, this study develops an integrated framework for enterprise profiling based on text information extraction and industry classification. The framework links preprocessing, feature engineering, classifier comparison, and ensemble inference within a single workflow, and the experiments show that the proposed combination of methods produces strong overall accuracy while improving recognition for minority categories. The results also suggest that the approach is useful for practical tasks such as supply-chain partner identification, industrial analysis, and enterprise intelligence services. At the same time, several limitations should be acknowledged. The current dataset still has limited coverage, the analysis relies mainly on textual evidence rather than richer relational or multimodal signals, and large pretrained models continue to impose nontrivial computational costs. Future work can therefore explore multimodal fusion, deeper integration with enterprise knowledge graphs, assistance from larger language models, and lightweight deployment strategies so that the system becomes more accurate, more interpretable, and easier to apply in real settings.

References

[1] Singh, S. (2018). Natural language processing for information extraction. *arXiv preprint arXiv:1807.02383*.
[2] Devlin, J., Chang, M. W., Lee, K., & et al. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. *Proceedings of NAACL-HLT*, 4171-4186.
[3] Vaswani, A., Shazeer, N., Parmar, N., & et al. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, 5998-6008.
[4] Mikolov, T., Chen, K., Corrado, G., & et al. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.
[5] Pennington, J., Socher, R., & Manning, C. D. (2014). GloVe: Global vectors for word representation. *Proceedings of EMNLP*, 1532-1543.
[6] Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning, 20*(3), 273–297.
[7] Breiman, L. (2001). Random forests. *Machine Learning, 45*(1), 5–32.
[8] Kim, Y. (2014). Convolutional neural networks for sentence classification. In *Proceedings of EMNLP* (pp. 1746–1751).
[9] Chawla, N. V., Bowyer, K. W., Hall, L. O., et al. (2002). SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research, 16*, 321–357.
[10] Lin, T. Y., Goyal, P., Girshick, R., et al. (2017). Focal loss for dense object detection. In *Proceedings of ICCV* (pp. 2980–2988).