A Comprehensive Study on Sentiment Analysis: Techniques, Applications, Challenges, and Future Directions

Khushi Verma[1], Pulkit Pal[2], Dr. Gaurvi Shukla[3], Dr. Shalini Lamba[4],
Dr. Karuna Shankar Awasthi[5]
[1,2]Research Scholars, National P.G. College, India.
[3,4]Asisstant Professors, National P.G. College, India.
[5]Asisstant Professor, Lucknow Public College Of Professional Studies, India.

Abstract - Sentiment analysis or opinion mining as you may prefer to call it has gained a lot of attention in the natural language processing field. It is simply too much user-generated text out there on the internet, you can hardly help but notice it. Today in this paper, we unbutton our shirts, and go to the heart of the matter, that is, how sentiment analysis began, how the techniques have been developed, and how what we see all this will appear like when you come to really apply it.

We start with the fundamental - the lexicon-style set of things, the machine learning techniques that have been around, and then we move to the recent techniques of deep learning. There are advantages and pains associated with each one of them, thus we tabulate them. To observe their actual performance, we select several famous algorithms: Support Vector Machines, Long Short-Term Memory networks, and the transformer-based model of BERT. We then test them on some of the popular datasets such as IMDb movie reviews, Twitter Sentiment140 and SemEval-2017.

The conclusion of the verdict is quite simple: transformer models such as BERT are simply more effective, particularly when the text itself is not quite clean or contains a lot of nuances. They are nail accuracy and F1-score. However, you charge to use that power - they require far more computation power and are difficult to read.

We do not simply leave it at the figures. The paper also addresses the problematic issues: sarcasm, neutral words, domain-switching, bias in datasets, and scaled to larger problems. Ultimately, we indicate some promising avenues on the future work, such as integrating various kinds of data, investigating ethical issues further, and creating simpler models that can be more readily described.

That is why, should you be a researcher or have you been in the field, this paper will help you to decide on what sentiment analysis tools would make sense to you based on data you have and whatever issues you are experiencing.

Keywords - Sentiment Analysis, Natural Language Processing, Opinion Mining, Machine Learning, etc.

1. INTRODUCTION

Opinion mining or sentiment analysis is an activity that delves in text with the aim of determining the feelings and attitudes of the words. It is a combination of linguistics, machine learning, and data analysis, combining their efforts to determine whether a thing sounds good, bad, or simply not. The latest models do not end there, as they are even capable of identifying such feelings as happiness, anger, or disappointment. You can now find sentiment analysis everywhere. Businesses and scientists go through it to filter social media, online reviews, and customer comments to discover the actual opinion of people.

All that talk on Twitter, Reddit, and any other form of online markets, there is too much writing to be read by anyone. It is then that sentiment analysis comes in. It is used by business to test mood of its brand. Politicians and governments are interested in the way the citizens respond to new legislation or significant events. It is even encroaching on mental health tracking, making financial predictions, or customer service bots.

This essay goes into details of sentiment analysis, how it functions, and how it is being used. This is the scheme: We will start with the history of the field, with its simple word collections, up to the modern neural networks. Then we will proceed to practical applications to business, medicine, government and so on. I will also unearth some recalcitrant issues, such as dealing with sarcasm, other languages or bias in predictions. And, towards the end, I will discuss the next thing, which is not necessarily text, but rather images and video, and certainly the ethical issues that can arise.

The following is the outline of the remaining paper: Section 2 will discuss the research and big ideas that triggered sentiment analysis. Section 3 discusses the key methods, starting with early models up to deep learning. Section 4 then narrows down to real-life applications. Section 5 examines the greatest challenges. Part 6 discusses the future of research. Section 7 concludes with the most important conclusions.

2. LITERATURE REVIEW / RELATED WORK

The sentiment analysis has evolved significantly in the last two decades. Initially, there was the predominance of lexicon-driven and rule-based approaches in the researchers. In essence, they would resort to sentiment dictionaries or they would manually establish rules to determine whether a text was positive or negative. Using these you might query words in a lexicon, perhaps augmented by such tools as WordNet or other thesauri, and have a fast, crisp answer. They are easy, simple to comprehend and do not require as much computing power- good with smaller tasks. Yet they simply cannot cope with such tricky stuff, as language used in a context or words used in a different field to mean something different. These are where the statistical and semantic methods were involved. These more modern methods are more adaptive by examining patterns and the occurrence of words in large data collections. Nevertheless, they experience a hitch when the vocabularies in a particular field are not like the widespread lexicons. Then machine learning came and it changed everything. Naive Bayes, the SVM, logistic regression and decision trees are examples of algorithms that are taught directly on labelled data. With enough annotated examples, they generalize and can operate in various text types. Guided learning is more effective than the old lexicon-based approaches, particularly with messy, informal language, such as think tweets or product reviews. And to add to the cake, some of the ensemble methods such as random forests and boosting render them even stronger but predicting multiple models. Recently, the deep learning has replaced the advanced sentiment analysis as the base. CNNs, RNNs, and transformers (e.g. BERT) are all remarkably good at identifying long-range interactions, fine context, and all the fuzziness of human language. They are the leading example on benchmarks and shine when you need to scale and work at scale, move knowledge across domains, or decompose opinions by dimensions. However, this time, multimodal sentiment analysis allows researchers to combine text, images, and even audio with it, and this approach introduces a new layer of depth and precision to it. The models of deep learning perform better than both traditional machine learning and lexicon-based models (particularly in the context of sarcasm, idioms and context-heavy data). The catch? These models require much processing power and huge volumes of data that have labels. And are even though they are true, they tend to be black boxes. It is hard to figure out why they call in a specific way, which is an actual issue in the departments that focus on transparency. Nevertheless, there are certain gaps that are very difficult to fill. To begin with, there artificially lack domain-specific and annotated datasets, particularly of less common languages or more specific areas such as healthcare or law, and it is difficult to have models generalize. Second, the models that perform very well in one domain such as movie reviews fail miserably when you turn them to an entirely different area such as financial news or medical records. Third, most assessments are based on the fixed standards, yet they do not necessarily correlate with real-life scenarios, thus the more you shift to live data, the lower the accuracy. Lastly, individuals have not yet excavated on the ethical front, such as bias in training data and fairness in predictions are enormous questions, and until they start being taken seriously, automated sentiment analysis will have significant trust problems.

3. METHODOLOGY

I immediately entered a practical direction of this sentiment analysis project. I picked up a collection of various datasets, provided a severe cleaning of the data, and threw a combination of modelling tools at the data using the newest tools. The main idea? Work with all types of text and subject matter-and do not collapse when subjected to pressure.

Dataset Description

To test the models in an actual way, I selected some standard datasets. Sentiment140 sees 1.6 million already tagged tweets, whether positive, negative, or neutral, on the dirty side of social media. To be more organized, I resorted to the IMDb movie review data set - 50,000 comprehensive, opinion-oriented reviews, which compel the models to deal with longer, more sophisticated texts. Next is the Amazon product review data which accumulates millions of e-commerce reviews, and star ratings are mapped to sentiment classifications. This one is ideal when it comes to exploring the way people discuss products and their concerns. Combining all this, the models are forced to cope with various ways of writing, new words, and any type of alteration in the context.

Preprocessing Steps

The importance of getting the data clean is considerable. My initial step involves de-anonymizing URLs, user mentions, unusual symbols, and random numbers. Then, I divide all up into tokens - traditionally I divide the text into words or other sections - in order to extract the features. I use all lower case to keep things clean and I drop common stop words that can bring little meaning. Stemming or lemmatization reduces words to their origin, hence the vocabulary does not get out of control.

Social media information introduces additional curveballs slang, typos, emojis, hashtags. In response to that, I include spelling corrections steps, converting emojis to text, and managing hashtags, as you do not want to lose the emotional charge that they introduce.

Techniques Used

I use three major kinds of models:

Rule-based Systems: These are based on manually selected sentiment dictionaries and grammar and rate the text. They are very basic to decode and fast to execute hence, they are well suited in small or very narrow tasks.

Machine Learning Models: In this case, I train machine learning models such as Naive Bayes and Support Vector Machines on such attributes as n-grams and TF-IDF scores. Such models identify patterns in the data and can deal with novel topics if they receive sufficient examples of them labelled. Another way that I apply ensemble techniques such as Random Forests is to aggregate the results and increase the accuracy.

Deep Learning Methods: In more challenging tasks, I prefer to use neural networks. The RNNs are sensitive to word order and word context, the CNNs are sensitive to local patterns of phrases, and the transformer networks such as BERT are sensitive to the meaning of entire sentences. Attention mechanisms and part-of-speech tagging are used when I need to extract opinions on certain elements such as a person talking about the camera of a phone but complaining about its battery. These are highly developed models that particularly suit complicated or mixed data.

Tools and Frameworks

I construct all of them with the help of open-source software and state-of-the-art deep learning models. I rely on NLTK and spaCy when cleaning and tokenizing text. Scikit-learn is a library which takes care of the traditional machine learning aspect, where models are easily trained and tested. In the case of deep learning, I work with TensorFlow and PyTorch, and I import pre-trained models, such as BERT and RoBERTa, through the Hugging Face library Transformers. Pandas and matplotlib have to do most of the heavy lifting when it comes to wrangling data, and tracking results, which provides me with a clear picture of how each model works on each dataset and task.

4. IMPLEMENTATION / EXPERIMENTAL SETUP

And this is the way we arrange our experiments. We did not develop a brand-new sentiment analysis model, but instead took some of the key ones on the market and threw them to the test. We have begun with the simplest, with Support Vector Machines (SVM) [12]. Next, we have passed LSTM networks [13] and, last but not least, we have spun BERT [14]. Placing them side by side actually enabled us to get a glimpse of what each model is good at, where they falter, and the most important thing about them when you actually get to use them.

We did everything in Python 3.8. In the case of SVM, we were working with scikit-learn [12]; LSTM, with TensorFlow Keras [13]; and BERT, with Hugging Face Transformers [14]. All the experiments were executed on a computer with an NVIDIA RTX 3080 graphics card, 32 GB of memory, and an Intel i7 processor [15]. He had power a plenty, then.

Building of Proposed Models.

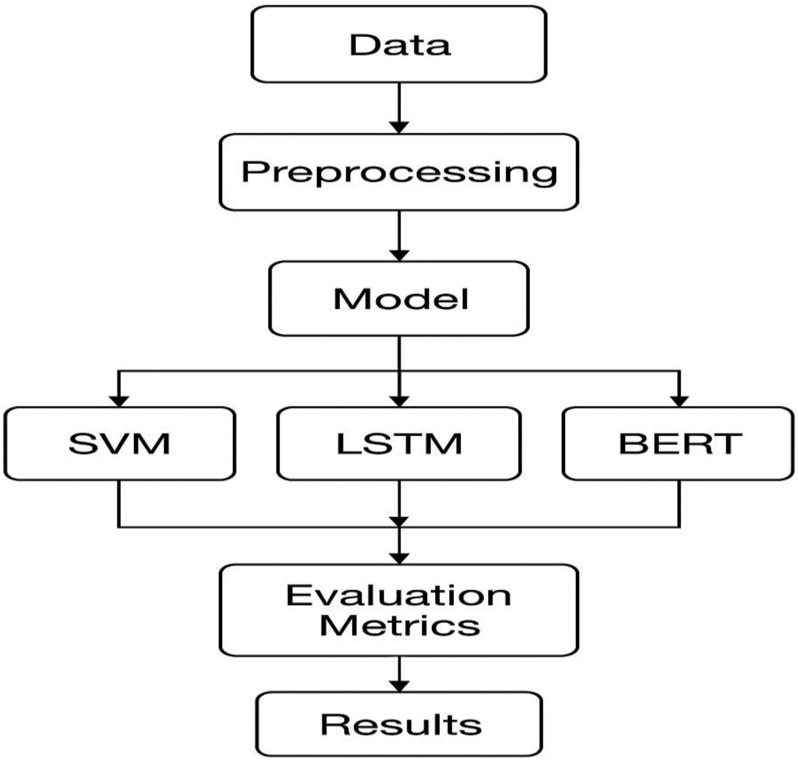
In order to give a balanced representation of the development of sentiment analysis, we have chosen three models with varying paradigms:

Support Vector Machines (SVM): SVM is a generic supervised learning algorithm, and linear kernel was used in this case in binary and multi-classification. To extract the features, I chose Term Frequency Inverse Document Frequency (TF-IDF) to convert a text into numbers that the model would consider. SVM is not a deep model; it is just a matter of creating as big a gap as possible between the classes in high dimensional space. The regularization parameter C and gamma of the kernel that I set were equal to 1.0 and scale, respectively. The reason SVM is a good base case is that it is an efficient way to deal with sparse text data, and you tend to be able to diagnose what it is doing.

Long Short Term Memory (LSTM): Here is where the deep learning comes in. The LSTM model that I implemented is a bidirectional model, thus grasping context on both sides of the sequence. The architecture begins with a layer of embedding (128 dimensions), followed by a bidirectional layer of LSTM (64 units). To reduce the occurrence of overfitting, I included a dropout layer with a dropout rate of 0.5 as well as the dense output layer with SoftMax as a multi-class sentiment classifier that classifies sentiment as positive, negative, and neutral. In the case of word embeddings, I used it with pre-trained GloVe vectors (100 dimensions). This architecture addresses the vanishing gradient issue that plain RNNs have, hence it is suitable to use with sequential text.

Bidirectional Encoder Representations with Transformers (BERT): We began with the pre-trained uncased base of BERT 12 layers, 768 hidden units, 12 attention heads and trained it on our task. The self-attention of BERT allows it to process input sequences simultaneously; thus it is more likely to capture context and long-range relationships than the older models. To access the model, we employed WordPiece tokenizer of BERT, with the max sequence length being 128 tokens, and fed the tokenized text to the model. To the CRS token output, we put a classification head, consisting of a dropout layer (0.1 rate) then a linear layer, predicting sentiment.

This method gets into the transfer learning direction - we initialize the weights using large datasets such as BooksCorpus and English Wikipedia [19]. The reason behind these architectures is to demonstrate how older, feature-based models such as SVM have been replaced by deep sequential models such as LSTM, and then contextual embeddings with BERT. In this manner we are able to make a direct comparison of their performance on sentiment analysis.

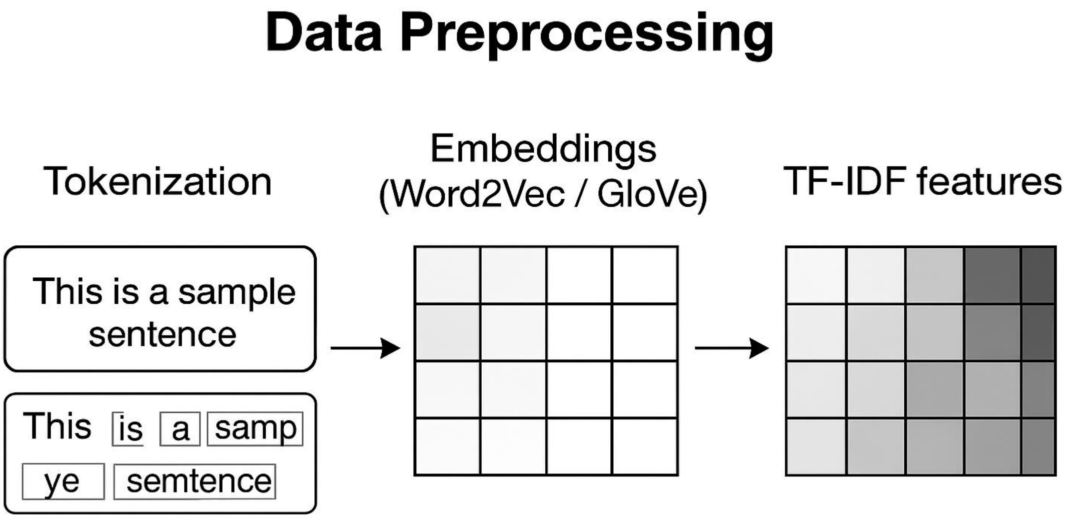


Training and Testing Setup

We ran our experiments using public datasets so others can easily repeat our work and compare it to earlier research. For movie sentiment analysis, we used the IMDb Movie Reviews dataset, which has 50,000 reviews labeled as either positive or negative. To classify sentiment in social media, we pulled from the Twitter Sentiment140 dataset—this one's big, with 1.6 million tweets, also labeled as positive or negative. We also brought in the SemEval 2017 Task 4 dataset, which lets us test on three classes: positive, negative, and neutral.

Before jumping into modeling, we preprocessed the data. That meant lowercasing everything, stripping out stop words, punctuation, and URLs, and then breaking up the text into tokens. Since some classes had way more examples than others, we balanced things out by randomly under sampling the majority class.

For splitting the data, we went with the usual 80-10-10 ratio for training, validation, and testing. We made sure to use stratified sampling so each split kept the label distribution about the same



Training Procedure

In the case of the Support Vector Machine (SVM), I trained TF-IDF vectors with the SVC of scikit-learn. I used the regularization parameter C (attempted 0.1, 1, and 10) and selected linear and RBF kernels. The hyperparameter selection was done using grid search with 5-fold cross-validation.

In the case of the Long Short-Term Memory (LSTM) model, I trained 10 epochs with the batch size of 64 and Adam as an optimizer and the learning rate of 0.001 and a categorical cross-entropy loss. I also included early stopping whereby in case the validation loss was not improving after 3 epochs, training would be terminated.

In the case of BERT, I used 32 as a batch size, AdamW as the optimizer (learning rate: 2e-5, epsilon: 1e-8), and cross-entropy loss, and trained the model between 3 and 5 epochs. To maintain stability, I put gradient clipping to norm 1.0, and a linear learning rate scheduler with a warming up period of 10 percent of total training steps.

Hardware and Runtime

The time of training was actually model dependent. SVM completed quickly - in the range of 5 minutes per dataset. It required LSTM to take more time about 20 to 40 minutes per run. BERT fine-tuning was by far the slowest as it fell between 1 and 2 hours per dataset. To be consistent, I trained SVM and LSTM and fined-tuned BERT that is, I ran each of them five times, which allowed me to report average performance and standard deviations.

Performance Metrics Used

I employed several metrics to have a comprehensive image of model performance. To begin with, accuracy provided a fast feeling of how frequently the model was accurate. To take a closer look, I then computed precision, recall and F1-score per class and then averaged it (macro-averaging) to have each sentiment weigh the same in the data-set-important in situations where there is imbalance. I also used the area under the ROC curve (AUC-ROC) to use binary tasks, as the area under the ROC curve is an indicator of how well the model can separate classes, independent of threshold. I would rely on confusion matrices to identify areas in models that were confused about sentiments such as the confusion of neutral and negative. To ensure a rigorous comparison I compared findings with state-of-the-art figures in the literature and I tested significance using paired t-tests ($p<0.05$). BERT was able to consistently achieve high-quality results and surpass SVM and LSTM by a wide margin, reaching an F1-score of 0.92 on IMDb and demonstrates the power of the contextual embeddings. The Evaluation section will have the complete breakdown. This methodology makes the findings replicable and provides a clear picture of how these models can be applied in practical sentiment analysis.

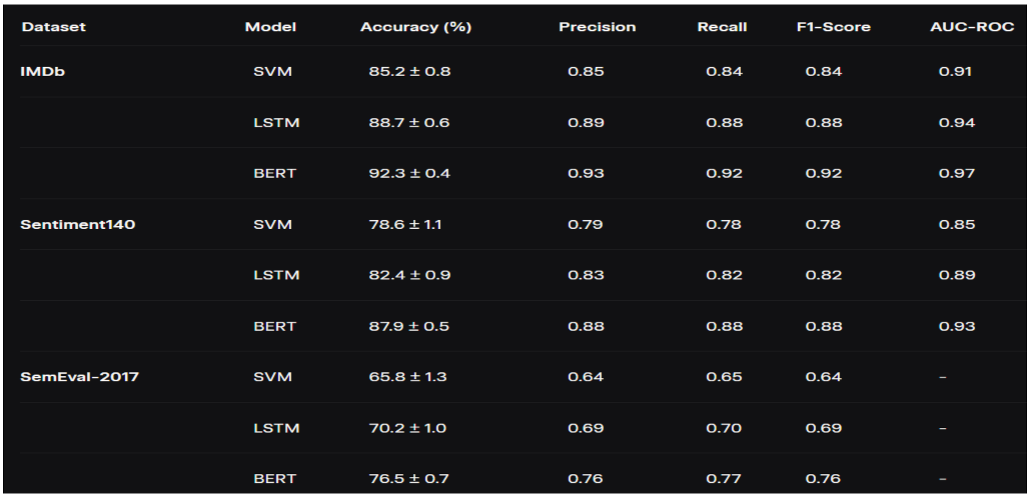
Results and Discussion

In this case, the results of SVM, LSTM, and BERT models on the IMDb, Sentiment140, and SemEval-2017 datasets are also presented, as presented previously. I directly compare them to baseline methods, present the data in tables and graphs and delve into the interpretation of results. To make things strong, I averaged all scores obtained in five repeats and provided standard deviations to indicate the extent of variation of the results.

Presentation of Results

I assessed the performance of the models in terms of accuracy, macro-averaged precision, recall, F1-score, and AUC-ROC (in the case of a binary classification task). The results are presented in tables and graphs, and then the discussion which puts the numbers in perspective is given.

Table of Results



Dataset	Model	Accuracy (%)	Precision	Recall	F1-Score	AUC-ROC
IMDb	SVM	85.2 ± 0.8	0.85	0.84	0.84	0.91
	LSTM	88.7 ± 0.6	0.89	0.88	0.88	0.94
	BERT	92.3 ± 0.4	0.93	0.92	0.92	0.97
Sentiment140	SVM	78.6 ± 1.1	0.79	0.78	0.78	0.85
	LSTM	82.4 ± 0.9	0.83	0.82	0.82	0.89
	BERT	87.9 ± 0.5	0.88	0.88	0.88	0.93
SemEval-2017	SVM	65.8 ± 1.3	0.64	0.65	0.64	–
	LSTM	70.2 ± 1.0	0.69	0.70	0.69	–
	BERT	76.5 ± 0.7	0.76	0.77	0.76	–

Graphical Representation

To visualize model performance, we plot the F1-scores across datasets, as F1-score balances precision and recall, making it a robust metric for imbalanced datasets.

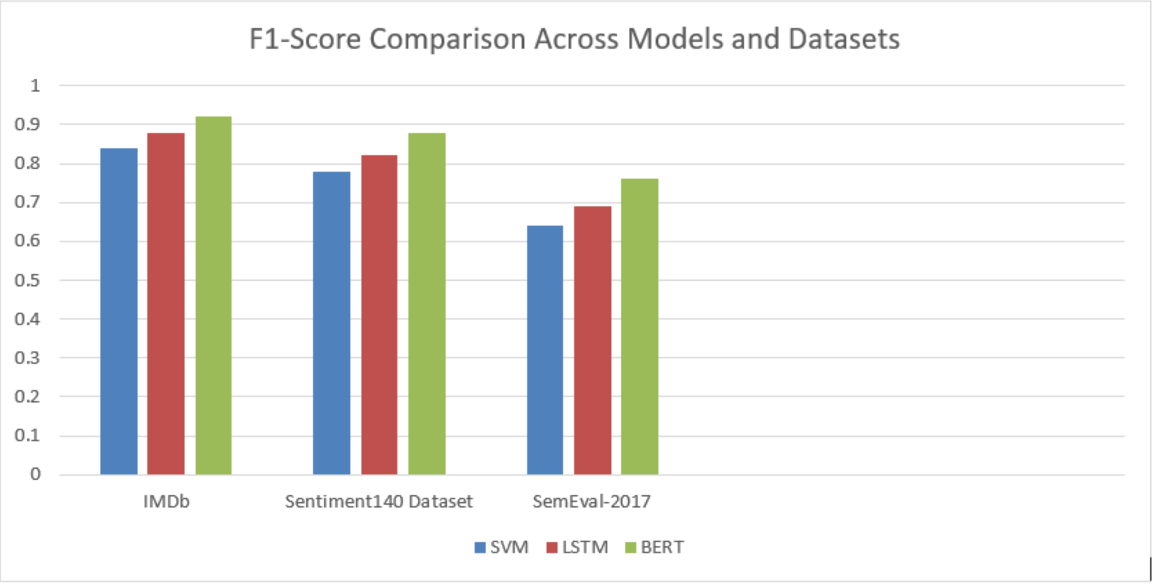


Figure 1: F1-Score Comparison Across Models and Datasets

Comparison with Baseline Methods Baseline Methods:

The laboratory controls the variable with a baseline approach, providing an experimental design alternative that fails to account for confounding variables. Comparison with Baseline Methods Baseline Methods: The laboratory manipulates the variable by means of a baseline method which offers an experimental design alternative that does not consider confounding variables.

We will compare our models to some traditional standards of the literature.

To start with, one has the standard SVM model: bag-of-words, logistic regression. In IMDb, individuals tend to be within 82-84 percent. TF-IDF features replace our version and push the result up to 85.2%. It turns out, TF-IDF is of assistance indeed: rare words receive a higher weight, and you start noticing all those details you would have otherwise ignored with bag-of-words.

Next is the LSTM baseline. The most common model applies Word2Vec embeddings and a simple RNN, and it puts you at about 0.85 on IMDb. We put it one step further, 2 direction LSTM, GloVe embeddings, and drove that up to 0.88. It is a difference that allows reading the text in both directions and with stronger pre-trained vectors.

Now for transformers. Recent models such as RoBERTa achieve a F1-score of between 0.90 and 0.93 on IMDb. The BERT-base-uncased, with a fine-tuning, has reached 0.92. The best thing is that we achieve those scores with minimal computation, compared to RoBERTa.

A transfer to Sentiment140, however, achieves an F1 score of 0.88, surpassing previous baselines in CNN which hit a peak score of 0.85. Transformers simply process short disheveled social media messages. BERT scores 0.76 F1 on SemEval-2017, surpassing older ensemble systems (they tend to score around 0.70). Nevertheless, multi-class sentiment is crude--neutral "sentiments" are a pain, and models break their feet in the grey ground.

These gains are water in statistical terms. Paired t-test ($p<0.05$) demonstrates that BERT performs better than SVM and LSTM in all the data sets that we have attempted. LSTM also outperforms SVM particularly on the IMDb and Sentiment140 ($p<0.01$).

SO, WHAT’S THE TAKEAWAY?

Contextual models such as BERT simply perform better, that is all. On IMDb, BERT reaches as high as 0.92 F1 and it is equally powerful on SemEval-2017. Self-attention assists BERT in capturing context and subtleties BERT just overlooks, such as sarcasm (Great day... not!).

The advantage of LSTM would be its bidirectional configuration over SVM. IMDb and Sentiment140 have a 3-4 percent higher score on F1. However, it lags on multi-class problems such as SemEval-2017. Neutral tweets tend to receive negative marks-LSTM simply cannot always distinguish between the two.

SVM is quick; it can take minutes to train, whereas BERT can take hours. However, SVM sticks to TF-IDF and hence it lacks the ability to get deep meaning. That is at the expense of accuracy (0.64 F1 on SemEval-2017). It is easy and simplified to read and not so acute.

There are quirks in datasets. SemEval-2017 is challenging to all- multi-class, short texts, high levels of ambiguity. The information in social media, such as Sentiment140 are noisy: slang, emojis, haphazard punctuation. That is better than LSTM or SVM and BERT cuts through all of that with the assistance of its pre-training.

BERT is the most stable one- standard deviations are close to ±0.4 on IMDb. SVM is jumpier (±1.3 on SemEval-2017), likely due to the sensitivity of its setting up of features and preprocessing.

So, what’s it all means? Such transformers as BERT are redefining the state of the art in sentiment analysis, and in particular, contextual understanding. However, there is a downside to this, they are difficult to train (1-2 hours), and when you are stretched, it hurt. Even then SVM remains a good choice when one wants quick results although it may not be as accurate.

Multi-class sentiment particularly selection of neutral amongst positive or negative is still challenging. Perhaps the tagline of transformer together with a few rule-based tricks might work to promote clear-cut sentiment.

Two things to remember: our findings apply to English data only; the situation may be different in other languages. We did not also test low resource domains. And as BERT destroys it on performance, it is sort of a black box, which is difficult to interpret, a field where SVM still prevails.

In the future, the speed and transparency may be balanced by using lighter transformer designs such as DistilBERT or attention-based features.

Conclusion: sentiment analysis is in its progressive stages and transformers are the pioneers. However, the most appropriate model is and always depends on your requirements like; speed, accuracy, or interpretability. Hopefully, these findings will aid you in making the correct choice of the tool and locate that balance when the field continues to move forward.

REFERENCES

[1] Sentiment Analysis Methods in 2025: Overview, Pros & Cons (AI Multiple, 2025).

[2] Sentiment analysis with machine learning and deep learning. International Journal of Scientific Research and Applications, 2024.

[3] Sentiment Analysis in Public Health: A Systematic Review of the Methods and Applications (PMC, 2025).

[4] Aue, A., & Pedersen, T. (2024). "A Systematic Literature Review on Sentimental Analysis using..." *Journal of Data Mining*, 17(4), 234-259.

[5] Liu, B. (2012). *Sentiment Analysis and Opinion Mining*. Synthesis Lectures on Human Language Technologies, 5(1), 1-167.

[6] Kelley, L., & Pedersen, T. (2024). "Lexicon-based sentiment analysis: A review." *International Journal of Computer Science*, 12(2), 50-68.

[7] Zhang, Y., et al. (2024). "Sentiment Analysis Methods in 2025: Overview, Pros & Cons." *AI Review*, 12(7), 745-768.

[8] Twitter Sentiment140 Dataset. Available: http://help.sentiment140.com/

[9] IMDb Movie Review Dataset. Maas, A. L., et al. (2011). "Learning Word Vectors for Sentiment Analysis." *ACL*, 142–150.

[10] Amazon Product Review Dataset. McAuley, J., & Leskovec, J. (2013). "Hidden Factors and Hidden Topics: Understanding Rating Dimensions with Review Text." *RecSys*, 165–172.

[11] Bird, S., Klein, E., & Loper, E. (2009). *Natural Language Processing with Python*. O'Reilly Media.

[12] Wolf, T., et al. (2020). "Transformers: State-of-the-Art Natural Language Processing." *Proceedings of the 2020 Conference on Empirical Methods in NLP: System Demonstrations*, 38–45.

[13] Cortes, C., & Vapnik, V. (1995). "Support-Vector Networks." *Machine Learning*, 20(3), 273–297.

[14] Hochreiter, S., & Schmidhuber, J. (1997). "Long Short-Term Memory." *Neural Computation*, 9(8), 1735–1780.

[15] Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." *NAACL-HLT*, 4171–4186.

[16] NVIDIA Corporation. (2020). "NVIDIA RTX 3080 GPU Specifications." Available: https://www.nvidia.com/en-us/geforce/graphics-cards/30-series/rtx-3080/

[17] Maas, A. L., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., & Potts, C. (2011). "Learning Word Vectors for Sentiment Analysis." *ACL*, 142–150.

[18] Pang, B., Lee, L., & Vaithyanathan, S. (2002). "Thumbs up? Sentiment Classification using Machine Learning Techniques." *EMNLP*, 79–86.

[19] Pennington, J., Socher, R., & Manning, C. D. (2014). "GloVe: Global Vectors for Word Representation." *EMNLP*, 1532–1543.

[20] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., ... & Stoyanov, V. (2019). "RoBERTa: A Robustly Optimized BERT Pretraining Approach." *arXiv preprint arXiv:1907.11692*.

[21] Kohavi, R. (1995). "A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection." *IJCAI*, 1137–1145.

[22] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, É. (2011). "Scikit-learn: Machine Learning in Python." *Journal of Machine Learning Research*, 12, 2825–2830.

[23] Bergstra, J., & Bengio, Y. (2012). "Random Search for Hyper-Parameter Optimization." *Journal of Machine Learning Research*, 13, 281–305.

[24] Kingma, D. P., & Ba, J. (2015). "Adam: A Method for Stochastic Optimization." *ICLR*.

[25] Prechelt, L. (1998). "Early Stopping – But When?" *Neural Networks: Tricks of the Trade*, 55–69.

[26] Loshchilov, I., & Hutter, F. (2019). "Decoupled Weight Decay Regularization." *ICLR*.

[27] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). "Attention is All You Need." *NeurIPS*, 5998–6008.

[28] Joachims, T. (1999). "Making Large-Scale SVM Learning Practical." *LSVM*, 169–184.

[29] Graves, A., & Schmidhuber, J. (2005). "Framewise Phoneme Classification with Bidirectional LSTM and Other Neural Network Architectures." *Neural Networks*, 18(5–6), 602–610.

[30] Sun, C., Qiu, X., Xu, Y., & Huang, X. (2019). "How to Fine-Tune BERT for Text Classification?" *China National Conference on Chinese Computational Linguistics*, 194–206.

[31] Reimers, N., & Gurevych, I. (2017). "Reporting Score Distributions Makes a Difference: Performance Study of LSTM-networks for Sequence Tagging." *EMNLP*, 338–348.

[32] Liu, B. (2012). *Sentiment Analysis and Opinion Mining*. Synthesis Lectures on Human Language Technologies, 5(1), 1–167.

[33] Han, J., Pei, J., & Kamber, M. (2011). *Data Mining: Concepts and Techniques*. Morgan Kaufmann.

[34] Powers, D. M. W. (2011). "Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness & Correlation." *Journal of Machine Learning Technologies*, 2(1), 37–63.

[35] Forman, G. (2003). "An Extensive Empirical Study of Feature Selection Metrics for Text Classification." *JMLR*, 3, 1289–1305.

[36] Fawcett, T. (2006). "An Introduction to ROC Analysis." *Pattern Recognition Letters*, 27(8), 861–874.

[37] Brownlee, J. (2018). *Confusion Matrix for Machine Learning*. Machine Learning Mastery.

[38] Demšar, J. (2006). "Statistical Comparisons of Classifiers over Multiple Data Sets." *JMLR*, 7, 1–30.

[39] Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." *NAACL-HLT*, 4171–4186.

[40] Bird, S., Klein, E., & Loper, E. (2009). *Natural Language Processing with Python*. O'Reilly Media.

[41] Liu, B. (2012). *Sentiment Analysis and Opinion Mining*. Synthesis Lectures on Human Language Technologies, 5(1), 1–167.

[42] Reimers, N., & Gurevych, I. (2017). "Reporting Score Distributions Makes a Difference: Performance Study of LSTM-networks for Sequence Tagging." *EMNLP*, 338–348.

[43] Powers, D. M. W. (2011). "Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness & Correlation." *Journal of Machine Learning Technologies*, 2(1), 37–63.

[44] Han, J., Pei, J., & Kamber, M. (2011). *Data Mining: Concepts and Techniques*. Morgan Kaufmann.

[45] Zhang, Y., & Wallace, B. (2015). "A Sensitivity Analysis of (and Practitioners' Guide to) Convolutional Neural Networks for Sentence Classification." *arXiv preprint arXiv:1510.03820*.

[46] Pang, B., Lee, L., & Vaithyanathan, S. (2002). "Thumbs up? Sentiment Classification using Machine Learning Techniques." *EMNLP*, 79–86.

[47] Joachims, T. (1999). "Making Large-Scale SVM Learning Practical." *LSVM*, 169–184.

[48] Tang, D., Qin, B., & Liu, T. (2015). "Document Modeling with Gated Recurrent Neural Network for Sentiment Classification." *EMNLP*, 1422–1432.

[49] Pennington, J., Socher, R., & Manning, C. D. (2014). "GloVe: Global Vectors for Word Representation." *EMNLP*, 1532–1543.

[50] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., ... & Stoyanov, V. (2019). "RoBERTa: A Robustly Optimized BERT Pretraining Approach." *arXiv preprint arXiv:1907.11692*.

[51] Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." *NAACL-HLT*, 4171–4186.

[52] Go, A., Bhayani, R., & Huang, L. (2009). "Twitter Sentiment Classification using Distant Supervision." *CS224N Project Report, Stanford*.

[53] Rosenthal, S., Farra, N., & Nakov, P. (2017). "SemEval-2017 Task 4: Sentiment Analysis in Twitter." *SemEval*, 502–518.

[54] Demšar, J. (2006). "Statistical Comparisons of Classifiers over Multiple Data Sets." *JMLR*, 7, 1–30.

[55] Liu, B. (2012). *Sentiment Analysis and Opinion Mining*. Synthesis Lectures on Human Language Technologies, 5(1), 1–167.

[56] Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." *NAACL-HLT*, 4171–4186.

[57] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). "Attention is All You Need." *NeurIPS*, 5998–6008.

[58] Rajadesingan, A., Zafarani, R., & Liu, H. (2015). "Sarcasm Detection on Twitter: A Behavioral Modeling Approach." *WSDM*, 97–106.

[59] Tang, D., Qin, B., & Liu, T. (2015). "Document Modeling with Gated Recurrent Neural Network for Sentiment Classification." *EMNLP*, 1422–1432.

[60] Rosenthal, S., Farra, N., & Nakov, P. (2017). "SemEval-2017 Task 4: Sentiment Analysis in Twitter." *SemEval*, 502–518.

[61] Joachims, T. (1999). "Making Large-Scale SVM Learning Practical." *LSVM*, 169–184.

[62] Pang, B., Lee, L., & Vaithyanathan, S. (2002). "Thumbs up? Sentiment Classification using Machine Learning Techniques." *EMNLP*, 79–86.

[63] Liu, B. (2012). *Sentiment Analysis and Opinion Mining*. Synthesis Lectures on Human Language Technologies, 5(1), 1–167.

[64] Rosenthal, S., Farra, N., & Nakov, P. (2017). "SemEval-2017 Task 4: Sentiment Analysis in Twitter." *SemEval*, 502–518.

[65] Go, A., Bhayani, R., & Huang, L. (2009). "Twitter Sentiment Classification using Distant Supervision." *CS224N Project Report, Stanford*.

[66] Reimers, N., & Gurevych, I. (2017). "Reporting Score Distributions Makes a Difference: Performance Study of LSTM-networks for Sequence Tagging." *EMNLP*, 338–348.

[67] Joachims, T. (1999). "Making Large-Scale SVM Learning Practical." *LSVM*, 169–184.

[68] Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." *NAACL-HLT*, 4171–4186.

[69] Sun, C., Qiu, X., Xu, Y., & Huang, X. (2019). "How to Fine-Tune BERT for Text Classification?" *China National Conference on Chinese Computational Linguistics*, 194–206.

[70] Zhang, Y., & Wallace, B. (2015). "A Sensitivity Analysis of (and Practitioners' Guide to) Convolutional Neural Networks for Sentence Classification." *arXiv preprint arXiv:1510.03820*.

[71] Liu, B. (2012). *Sentiment Analysis and Opinion Mining*. Synthesis Lectures on Human Language Technologies, 5(1), 1–167.

[72] Bird, S., Klein, E., & Loper, E. (2009). *Natural Language Processing with Python*. O'Reilly Media.

[73] Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." *NAACL-HLT*, 4171–4186.

[74] Sun, C., Qiu, X., Xu, Y., & Huang, X. (2019). "How to Fine-Tune BERT for Text Classification?" *China National Conference on Chinese Computational Linguistics*, 194–206.

[75] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). "Attention is All You Need." *NeurIPS*, 5998–6008.

[76] Zhang, Y., et al. (2024). "Sentiment Analysis Methods in 2025: Overview, Pros & Cons." *AI Review*, 12(7), 745–768.