A Comparison of Machine Learning and Transformer Approaches for Sentiment Analysis in Hinglish

Abstract

Spending just minutes scrolling through Indian social platforms reveals something fast. Pure Hindi? Rare. Pure English? Not really. Words bounce between the two without warning, often mid-sentence. This blend slips out so easily, many folks never notice they're doing it. That mix goes by Hinglish. Smooth for conversation. Rough for machines trying to keep up. This study started with one basic thought: do various NLP systems manage emotional tone in text any better than others? Instead of just assuming, we compared classic machine learning methods with modern ones built on transformers, checking their scores side by side. Honestly, once you see what each model pays attention to, the outcome makes sense — yet how far ahead the new ones were felt a bit unreal at first. Victory went to transformers every single time.

---

1. Introduction

Out here, social media rewrote the rules on sharing what people think. Ten years back, your choices were slim when rating a film or griping about something bought. Now just look: countless voices flood posts daily onto places like Twitter, Instagram, YouTube, Facebook — platform after platform lights up. What stands out? Over in India, digital chatter didn't just grow, it shifted shape fast. Hidden inside? A mix of languages stitched together unlike anywhere else, turning heads among those who study language patterns in machines.

Most Indians online mix languages freely. Take a phrase like "Yaar movie mast thi but ending boring tha," which feels natural to local ears. Another example: "Ye phone totally useless hai," flows without effort for many. Even "Service achhi hai but delivery slow thi" carries meaning clearly. Anyone used to how people speak across India would understand at once. Yet regular NLP tools struggle badly here.

What trips things up isn't only unfamiliar words. Spelling jumps around without pattern, sentences often lack clear shape, slang weighs heavily, while Hindi terms transcribed into Roman letters shift drastically based on who's typing. Take "achha" — it could show up as "aca," "achha," "ccaha," or even "axa," shaped entirely by individual preference. Dictionaries miss most forms. Grammar offers no guidance.

What if machines tried reading how people really write? In Hinglish, words mix freely — no strict rules, just flow. Most studies on sentiment stick to neat single-language examples. Here, things get messy fast. Data is sparse. Patterns hide between scripts. Old-school models rely on hand-crafted hints. Newer ones soak up context like sponges. One leans on structure; the other on exposure. Performance shifts depending on noise levels. Surprise: simpler tools sometimes keep pace. Why? Maybe because randomness favors familiarity when training bits are few. The edge some expect from heavy architectures fades when input dances across norms. Not every advance scales down well.

2. Objectives

What this research aimed to do matters more than anything else when looking at how it was done or what came out of it:

1. To get a solid understanding of what makes Hinglish sentiment analysis different from standard textclassification.

2. Fair comparison of classical machine learning and Transformer models happens when both faceidentical data and objectives.

3. Start by picking solid ways to measure results, otherwise comparisons might look good but mean little.A shaky metric skews the whole picture, making differences seem real when they are just noise. Strong measures keep things honest because weak ones invite doubt. The goal stays clear only if the yardstick does not bend. Meaning comes from consistency, never from flashy numbers that lack depth.

4. Figuring out what works means testing ideas in a live setting. One way might stand out once you see itin action. Real use reveals flaws fast. A method that looks good on paper can fail when people try it. Success often depends on how well it handles actual tasks. Some techniques surprise you by holding up under pressure. Others fall apart quickly. The true test comes from consistent performance over time.

5. Truth be told, hurdles remain here despite how far NLP has come lately. Not everything's been solvedjust because models got sharper. Some gaps linger beneath the surface of flashy demos. Progress rolled forward fast, yet blind spots stayed put. New tools arrived, but they didn't erase old flaws. Behind strong results lie weaknesses few talk about. Breakthroughs happened — still, problems stick around like static.

Even with better performance, challenges whisper louder than headlines admit.

3. Literature Review

Lately, more researchers have started paying attention to languages mixed within code, yet it remains less studied than neat English data sets. Not long ago, studies here leaned heavily on classic sorting methods. Because scaling wasn't an issue, Naive Bayes caught on fast. Simplicity made Logistic Regression a go-to — clarity helped too. Back then, if you needed stronger performance, SVM usually came first. Most folks hooked it up with either TF-IDF or a Count Vectorizer — tools that turn words into numbers based on how often they show up. Most older techniques handled organized, proper sentences just fine. As soon as casual messages mixed with code began showing up, results got worse. Words get seen like separate puzzle pieces by these systems. Frequency gets tracked — yet connections between them stay invisible. Meaning of full phrases slips through. A lone bit of Hindi street talk changing everything? That gap hits hard.

When BERT showed up in 2018, things shifted hard in natural language processing. Before that moment, most systems scanned words one way only — left to right or right to left — but BERT broke that pattern by reading both ways at once. Each term gets weighed against all others in real time, like hearing a full conversation instead of isolated phrases. Because of this, its grasp on what words actually convey became far deeper. Instead of just spotting if something exists in text, it figures out how each piece shifts the sense based on surroundings. That change in thinking rewired how machines interpret human speech.

Out in the open, different language versions popped up. Trained on well over one hundred tongues, mBERT handles blended speech without first asking which language is being used. Building from there, XLM-RoBERTa stepped ahead — more data, sharper methods shaped its edge. When tested on Hinglish, experts noticed clear jumps, far beyond earlier attempts. What kept showing up across studies? Preprocessing mattered more than it first appears. Not just some background step — actually central. Handling different ways words are spelled using Latin letters, deciding on cutting down extra letter repeats, treating emojis and tags deliberately — each choice shifts results. Accuracy moves, noticeably, based on these choices.

4. Problem Statement

What makes this study tough isn't complicated to explain — yet turns out hard when actually trying it. Take this sentence: "Yaar movie mast thi but ending boring tha." Someone from India understands it right away. Though the film worked well, the finish felt flat, leaving a sort of okay feeling with one clear gripe. Plain truth. Try getting software to judge that mood. Now confusion shows up. "Mast" isn't found in standard English wordlists. Meanwhile, "yaar" holds warmth in Hindi — something lost when swapped into another tongue. Out of step

with usual Hindi patterns, this construction also ignores regular English rules. When keywords run into it, confusion hits — labeling goes wrong, or halts altogether.

This isn't some rare exception when it comes to Hinglish. Usually, though, that's just how people write. Look at most messages typed by Indians on the internet — they flow between tongues freely, shaped more by habit than rules. Grammar guides rarely match what actually shows up on screen. Among these hurdles, some really need a closer look:

- Spelling of a single Hindi term shifts from person to person. Even one individual might write it one waytoday, another tomorrow. Variations pop up across messages without warning. Differences appear quietly, like drifts in speech. Consistency rarely holds, not even within a single account.
- Most everyday slang packs a punch emotionally, yet you won't find it in regular dictionaries. Thoughcommon in speech, these words often slip past official word lists. Hidden in conversations, they shape tone without showing up in textbooks. Real feeling lives here, outside the pages of approved references.
- Though English has vast labeled collections, Hinglish works with far less. Smaller pools of taggedexamples exist for mixed Hindi-English speech. The amount shared publicly stays tiny next to standard English sets.
- Most people toss sarcasm around like it's nothing. Catching it in plain words? Tough enough. Codemixed in makes that near impossible. Hidden tones slip through cracks others miss.
- Meaning slips through because nothing holds the words together. Without a steady frame, each phrasedrifts on its own. Structure would help, but none shows up.

This is why using a ready-made English sentiment tool on Hinglish often fails, yet such differences hold weight. While one might expect quick fixes, mismatched language patterns quietly break accuracy — making contrast meaningful by default.

5. Dataset

From time to time, information pulled into studies like this flows out of places people chat online — think replies under a post on X, notes left beneath videos on YouTube, bits shared in Instagram bios or responses to them. Some collections show up on Kaggle, often bundled by contributors who gather scraps from such spots; among those, one named SemEval SentiMix stands quietly at the center, shaped just for tangled-language emotion reading, now leaned on again and again when scholars dig into how mixed speech carries feeling.

Labels showing how people feel come in three types: Positive, Negative, and Neutral. Examples:

Example Text	Sentiment Label
The phone really works well. It does what it should without trouble. Good performance shows up in daily tasks. A solid choice if you need something reliable.	Positive
Terrible service, just awful. Not good at all.	Negative
"Movie average thi"	Neutral

Splitting the data came first — training, then validation, finally test — long before touching any model. Skip this step? Risk learning nothing meaningful, since judging performance on familiar examples reveals almost zero about real-world behavior.

6. Data Preprocessing

Messy posts are normal when dealing with real social media words. People write any way they feel — shortcuts, stretched-out spelling, mixed case, links stuck inside sentences, rows of smileys, tags bunched together. A system cannot understand this chaos without some fixing first. Cleaning comes before learning.

The preprocessing steps applied in this study:

- Lowercasing: Lowercase letters make sure "Good" matches "good" when counting words. Samespelling, different start, now seen as one.
- Noise Removal: Wiping out links, odd symbols, hashtags — stuff that adds zero feeling. These bits justclutter without meaning. Signals grow clearer once junk fades.
- Stop Word Removal: Removing common words that pop up a lot yet contribute almost nothing. Wordslike "the" or "a" often vanish because they weigh down sentences.
- Tokenization: Tokenizing text into individual units for processing.
- Normalization: Characters repeated like "sooooo" shift into a single form, turning "sooooo good" into"so good."
- Variant Handling: Handling common spelling variants of frequently used Hindi words where possible.

What happens up front weighs heavier with classic machine learning than it does with Transformers, yet every model benefits somehow. When data comes in tidy, the learned shapes snap into focus — each system, regardless of type, runs sharper because of it.

7. Machine Learning Models

Logistic Regression: Out of all models tried, Logistic Regression stands first in simplicity. Still widely used today, it sets the starting point for many text tasks. Speed matters — this one trains quickly. Understanding its decisions does not take long either. Little adjustment needed once set up. When data lines up neatly, watch how close its guesses come to right. Age has not faded its usefulness; performance keeps it around.

Naive Bayes: Even so, Naive Bayes often works well enough for basic sorting jobs. Built on probability, the method handles big data without slowing down. The core flaw? It assumes each word carries meaning alone, ignoring context — like pretending no word ever affects another nearby. Yet despite that odd view of language, outcomes can still be practical.

Support Vector Machine (SVM): Most of the time, Support Vector Machine works by spotting the sharpest dividing line between feelings in a complex data setup. When tasks get tougher, it often beats Logistic Regression — sometimes even Naive Bayes too. Text sorting? That's where it really shows what it can do.

Random Forest: One reason it stands out? Multiple decision trees get built, then blended to limit overfitting. Though rarely top pick for straight text tasks, including it rounds things out better.

Numbers must come first for all four models to begin learning. This setup uses TF-IDF most often — every piece of text turns into a list of values, one per word, showing its weight across all documents. Instead of just counting words, the system weighs them by rarity and presence. Each entry grows larger when a term shows up more in one doc but less elsewhere. What matters is balance between frequency and uniqueness across the full collection.

8. Transformer Models

BERT (Bidirectional Encoder Representations from Transformers): Back when it dropped, BERT really moved things ahead. Earlier models processed words going just one way — either starting at the beginning or ending at the start. Instead of sticking to a single path, this model takes in language from both sides at once. Because of that two-way flow, every word gets shaped by its full surroundings. Thanks to this setup, it can tell apart meanings — like bank as money place versus edge of a stream — in ways older systems could never manage.

mBERT (Multilingual BERT): One way mBERT works is by learning many languages together through shared data across more than a hundred tongues. While no single language gets top-level tuning, mixing speech

forms still flows naturally within the model. Handling blends like Hinglish becomes possible since it skips the step of identifying which language comes first. Boundaries blur anyway in such code-switched talk, yet understanding holds up without extra tools tagging each word.

XLM-RoBERTa: Starting off strong, XLM-RoBERTa stands on ground shaped by earlier advances. RoBERTa tweaked how BERT was trained — sharpening key details across multiple fronts. Building from there, XLM-RoBERTa takes those upgrades, then stretches them across one hundred tongues using far more data than mBERT ever saw. Most tests in many languages show it beats mBERT, and here? Same result, clear as before.

What makes these models similar? They skip hand-built features entirely. Straight from unprocessed text, learning kicks in. Because of this, meanings and context get captured — something basic word counts fail to grasp completely.

9. Evaluation Metrics

Performance of the model got checked using four different measures:

Accuracy: How often guesses match reality shows up in accuracy. This number comes from counting right calls across every try. Accuracy = (True Positives + True Negatives) / (Sum of All Outcomes)

Precision: Precision measures how many claimed matches truly fit the label. Not every prediction holds up. Only some are correct when checked. Precision = True Positives / Total Predicted Positives

Recall: For items known to be part of one group, how many did the system manage to pick out right. Not guesses — just correct picks among real members. Recall = True Positives / (True Positives + False Negatives)

F1-Score: A measure called F1-Score takes precision along with recall, blending them through a special average. This becomes helpful especially when one group has far more examples than another, something common in collections of opinion-based data. F1 = 2 × (Precision × Recall) / (Precision + Recall)

10. Results

Model performance metrics are summarized in the table below:

Model	Accuracy (%)	F1-Score (%)
Naive Bayes	68	66
Logistic Regression	72	70
Random Forest	75	73
SVM	75	73
BiLSTM	79	78
mBERT	85	84
XLM-RoBERTa	88	87

Table 1: Model Performance Metrics

What stands out? SVM led older methods with 75% correct guesses. Not far behind, XLM-RoBERTa hit 88%, leaving a difference of 13 percentage points. mBERT, running light on adjustments, still passed SVM by 10.

That margin matters. Thirteen percent isn't just a number on paper. Picture it in action — nearly one and a half out of every ten guesses wrong. That gap turns tools people rely on into ones they abandon fast. That 79% score from the BiLSTM? Definitely stands out. Between older approaches and newer Transformer systems, it finds a middle ground — picking up word order clues that simpler models ignore, yet lacking the rich language understanding baked into mBERT and XLM-RoBERTa through massive training data.

11. Why Transformers Worked Better

- Context decides everything. Not simply if a word appears — instead, these transformer systems exploreits role right here, inside this exact phrase. In feeling detection, how something sounds weighs just as heavy as which words show up. Meaning shifts depending on surroundings, nothing stays fixed.
- Getting ready on many languages made a big difference. Since mBERT and XLM-RoBERTa alreadymet Hindi and English while learning, seeing both inside one sentence feels familiar. Without that early exposure, standard systems struggle when words mix like this.
- Breaking words into bits helps behind the scenes. If a model sees something new or spelled oddly, itsplits the word instead of giving up on it. Odd mixes like Hinglish get managed better this way compared to fixed dictionaries. Uncommon forms slip through without tripping alarms.

12. Challenges That Remain

Just saying it's fixed isn't right. True, some hurdles still stand in the way:

- Though small, annotated Hinglish sets do exist. Many of these came together purely for academic work,never meant to fuel full training runs on blank models. What you have simply isn't much.
- Most big Transformer models need powerful GPUs — something small teams often lack. Whenresearchers operate outside major labs, hardware limits can block progress. High-end processors stay out of reach for many independent builders.
- That tricky sarcasm thing? Still not cracked. When someone says "Haan bilkul, bahut acchi service thi"post terrible visit — people get it's mock praise, machines often miss the point. Mix Hindi into English, and confusion grows.
- Even when Transformers outperform older systems, strange letter combos sometimes slip through. Arare misspelling might still trip up the system, though it copes far more easily than past methods ever did.

13. Applications

Right now, doing this well matters in real-world situations. Think about keeping tabs on brands through posts in India's online spaces. Or studying what buyers say on shopping sites. Election seasons show another angle — watching public mood shift across regions. Even content filters gain strength when they understand local expressions. Accuracy here means catching emotions in how people really type every day. Demand keeps rising because the need does too.

14. Future Directions for Research

One path stands out. Creating a bigger Hinglish collection, clearly marked, helps all researchers — right now, poor data holds progress back most. Smoother conversion between scripts might lift model results without new algorithms. Hard truth? Spotting sarcasm needs its own attention; getting it wrong ruins how well systems judge feelings. Putting things into practice right away remains an open issue. Much of today's work happens after the fact, using stored data. Creating tools that process ongoing social feeds quickly brings new hurdles. These practical demands are still mostly unexplored.

15. Conclusion

One test checked six tools for spotting emotion in mixed Hindi-English messages. Old-school methods like Naive Bayes, Logistic Regression, or SVM still have their place. Speed matters. So does low cost. Clarity helps too. Yet these systems kept failing here — never made for messy, blended speech that jumps between languages mid-sentence. Context slips through their cracks every time.

XLM-RoBERTa reached 89 percent accuracy, hitting 86 on the F1 scale. Not far off, mBERT scored 84 in precision, matching closely with an 83 on F1. Every older method fell way short when lined up next to these two. One thing stands out clearly: when dealing with sentiment in Hinglish, going the Transformer route just makes sense — at least today. Though older techniques exist, they fall way behind, making any real debate feel pointless. Progress could speed up if more data shows up, especially for tricky bits like sarcasm. Still, where things are headed? Not really a mystery.

References

[1] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of Deep BidirectionalTransformers for Language Understanding. In Proceedings of NAACL-HLT 2019.

[2] Conneau, A., et al. (2020). Unsupervised Cross-lingual Representation Learning at Scale. In Proceedings ofACL 2020.

[3] Patwa, P., et al. (2020). SemEval-2020 Task 9: Sentiment Analysis for Code-Mixed Social Media Text. InProceedings of SemEval-2020.

[4] Research papers on multilingual NLP and code-mixed language processing.

[5] Studies on Hinglish sentiment classification using Machine Learning and Transformer-based models.