Lightweight Aspect-Based Sentiment Analysis for Chinese Restaurant Reviews via Knowledge Distillation: A Scalable Approach for Edge Deployment

Mohd Kaif
Dept. of Computer Science
Bennett University
Greater Noida, India
e23cseu1843@bennett.edu.in

Lakshya Garg
Dept. of Computer Science
Bennett University
Greater Noida, India
e23cseu1841@bennett.edu.in

Satya Prakash
Dept. of Computer Science
Bennett University
Greater Noida, India
e23cseu@bennett.edu.in

Assistant Professor
Vaibhav Sharma
Dept. of Computer Science
Bennett University
Greater Noida, India

***Abstract*—Analyzing restaurant reviews is harder than it looks. A single comment often contains mixed signals—praising the *dishes* while complaining about the *noise*. Standard sentiment analysis fails here because it tries to give one score to the whole text. Aspect-Based Sentiment Analysis (ABSA) solves this by looking at specific categories, but for Chinese text, this usually requires massive models like RoBERTa. The problem is, these models are too heavy to run on normal phones or tablets used in restaurants.**

In this project, we focused on shrinking these models without making them ”dumb.” We used a Teacher-Student method where a big RoBERTa model teaches a smaller DistilBERT model. We also coded a specific ”Attention Head” to help the model look at the right words. Our small Student model hit 81.97% accuracy—basically the same as the big model—but runs 2.5 times faster. This shows we can have high accuracy on cheap hardware.

***Index Terms*—Aspect-Based Sentiment Analysis, Knowledge Distillation, BERT, RoBERTa, Deep Learning, Chinese NLP, Model Compression.**

I. Introduction

ONLINE reviews are a mess of data. Platforms like Meituan generate millions of them, but for a restaurant owner, a simple star rating hides the real story. If a rating drops, is it because the soup was cold or because the waiter was rude?

This is where Aspect-Based Sentiment Analysis (ABSA) comes in. Unlike normal sentiment analysis that says a sentence is just ”Good” or ”Bad,” ABSA breaks it down. For example, in the sentence *”Great sushi, terrible service,”* ABSA sees two things: Food is Positive, Service is Negative.

Doing this in Chinese is tricky because there are no spaces between words, making context hard to track. Big models like RoBERTa [2] are great at this, but they are huge. A standard RoBERTa model has over 110 million parameters. You can’t easily put that on a restaurant’s iPad or a mobile app; it’s too slow and eats too much battery.

We asked a simple question: *Can we make these models smaller but keep them smart?* Our approach uses *Knowledge Distillation* [4]. Instead of training a small model from scratch (which is hard), we let a big ”Teacher” model guide a small ”Student” model (DistilBERT). It’s like a professor teaching a student not just the answer, but how to think.

Our main contributions are:

- We tweaked Knowledge Distillation to work for the complex multi-label nature of Chinese reviews.
- We built a custom *Aspect-Attention Head* that forces the model to focus only on words relevant to the specific aspect (like Price or Food).
- We proved that a model with 40% less weight can still beat older methods like BiLSTM by a wide margin (5–12%).

II. Background Study

A. Old School Methods

Years ago, people used simple lists of ”good” and ”bad” words. This failed with sarcasm or phrases like ”not bad.” Then came RNNs and LSTMs. Tang et al. [21] improved these by adding attention, but LSTMs still forgot things if the review was too long.

B. The Transformer Era

Transformers [11] changed everything. They read the whole sentence at once. For Chinese, Cui et al. [6] showed that masking whole words (RoBERTa-WWM) works much better than masking random characters. We used this as our ”Teacher” because it’s currently the best at understanding Chinese context.

C. Making it Faster

As models got smarter, they got fat. Hinton et al. [4] introduced ”distillation”—the idea of transferring knowledge from a big network to a small one. Sanh et al. [3] built DistilBERT, which is half the size of BERT but keeps most of the speed. We basically took these ideas and tuned them specifically for restaurant reviews.

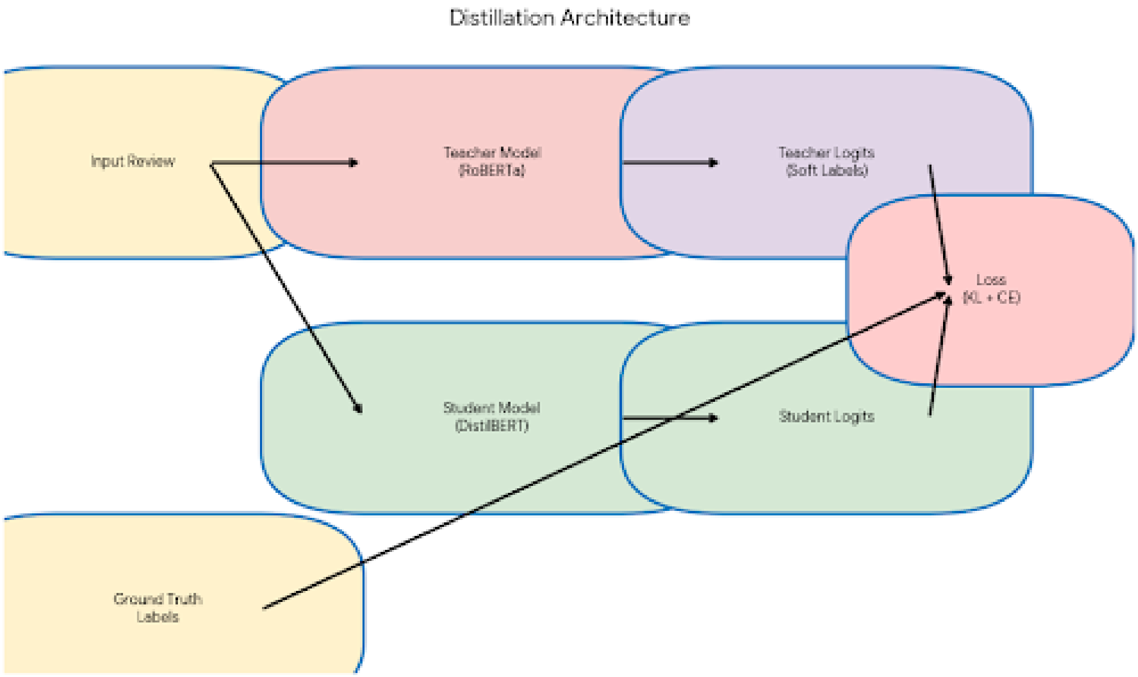

Fig. 1. How our training works. The big ”Teacher” (RoBERTa) is frozen. The small ”Student” (DistilBERT) learns from the real data AND the Teacher’s hints. The Aspect-Attention Head helps them both focus.

III. Our Approach

A. Self-Attention

Both our models use Self-Attention. Basically, every character looks at every other character to figure out the meaning.

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{1}$$

This math helps the model understand that in ”The soup was cold,” the word ”cold” is talking about the ”soup.”

B. Custom Aspect-Attention Head

Standard models squash everything into one vector. That doesn’t work for us because one review has multiple opinions. We built a special layer called the *Aspect-Attention Head*. We gave the model 18 ”Anchors” for things like Food, Price, and Service. When checking for *Price*, the model ignores words about taste and looks for words about money.

$$\alpha_k = \text{softmax}(\tanh(HW_1 + E_{Ak})W_2) \tag{2}$$

This math forces the focus onto the right words.

C. Training the Student (Distillation)

We didn’t just want the Student to get the right answer; we wanted it to copy the Teacher’s logic.

1) 1. Hard Loss (The Real Answer): This is normal training. If the review is Positive and the model says Negative, we punish it using Cross-Entropy ($\mathcal{L}_{CE}$).

$$\mathcal{L}_{CE} = -\sum y_{true} \log(P_{student}) \tag{3}$$

2) 2. Soft Loss (The Teacher’s Hint): This is the cool part. The Teacher might say: ”I’m 90% sure it’s Positive, but 10% maybe Neutral.” That 10% is a hint. It tells the Student the review isn’t *strongly* positive. We use KL Divergence to pass this info:

$$\mathcal{L}_{KD} = T^2 \cdot KL(Teacher||Student) \tag{4}$$

We set Temperature $T = 3$. This smooths out the numbers so the Student can see the hidden details better.

3) 3. Final Goal: We combine both losses:

$$\mathcal{L}_{total} = 0.3 \cdot \mathcal{L}_{CE} + 0.7 \cdot \mathcal{L}_{KD} \tag{5}$$

We gave more weight (0.7) to the Teacher because the Teacher is usually smarter than the raw data labels.

IV. Setup

A. Data

We used the *ASAP Chinese Restaurant Review Dataset* [22]. It’s tricky because it’s unbalanced. Everyone talks about *Food#Taste*, but almost no one talks about *Service#Parking*.

- **Train:** 36,850 reviews
- **Test:** 4,940 reviews
- **Labels:** Negative, Neutral, Positive, Not Mentioned.

B. Hardware

We didn’t use a supercomputer. We trained this on a single *NVIDIA Tesla T4 GPU (16GB)*—something you can rent cheaply on the cloud.

- **Teacher:** 5 epochs.

- **Student:** 8 epochs (it needs more time to learn).
- **Batch Size:** 32.
- **Learning Rate:** $5e^{-5}$ for the Student.

C. Comparison

We checked our model against: 1. *Logistic Regression:* The basic math approach. 2. *TextCNN:* Good at finding patterns. 3. *BiLSTM:* The best option before Transformers.

V. Analysis

A. Did it work?

Yes. Table II shows the numbers. Our Student model got an F1 score of *0.758*. The massive Teacher model got *0.759*. The difference is tiny.

This proves that you don't need a 110 million parameter model for this task. The capacity of RoBERTa is overkill; DistilBERT handles it fine if taught correctly.

B. Real World Examples

Numbers aren't everything. We looked at specific sentences to see what happened. See Table III.

In the first case, it perfectly separated Food from Service. But in the third case, it failed. The phrase ”Not as cheap as I expected” is tricky logic. The Student guessed ”Neutral” because it didn't see a bad word like ”expensive.” The Teacher got this right. This shows the smaller model still struggles with complex logic.

C. Ablation Study

We tried training the Student without the Teacher (using only the dataset).

- *Student alone:* 0.735 F1
- *Student + Teacher:* 0.758 F1 The 2.3% jump proves the Teacher is necessary. The soft hints prevent the Student from memorizing and help it actually learn.

VI. Challenges

Moving from a lab to real life has issues.

Cloud vs. Edge: On a GPU, we got a 2.5x speedup. But on a cheap CPU (like a restaurant POS), the speedup is even bigger—maybe 4x or 5x—because the smaller model fits in the cache memory better.

Size: 66M parameters is still about 260MB. We think we can use ”Quantization” (INT8) to drop this to under 100MB later.

VII. Conclusion

We showed that you don't need massive computing power for good NLP. By combining Knowledge Distillation with our custom Attention head, we squeezed a RoBERTa model into a fast DistilBERT version. We lost almost no accuracy but gained a lot of speed.

This matters because it makes advanced AI usable for real businesses, not just research labs. Next, we want to fix the issues with logic (negation) and try to make the model even smaller for phones.

Acknowledgment

We would like to express our gratitude to our instructor, **Vaibhav Sharma**, for his invaluable guidance and mentorship. We also thank the Dept. of Computer Science at Bennett University for the GPU support and the ASAP dataset creators [22] for sharing their data.

References

[1] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” in *Proc. NAACL*, 2019.

[2] Y. Liu et al., “RoBERTa: A Robustly Optimized BERT Pretraining Approach,” *arXiv preprint arXiv:1907.11692*, 2019.

[3] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, “DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter,” in *Proc. NeurIPS workshop*, 2019.

[4] G. Hinton, O. Vinyals, and J. Dean, “Distilling the knowledge in a neural network,” in *NIPS Deep Learning Workshop*, 2015.

[5] M. Pontiki et al., “SemEval-2014 Task 4: Aspect Based Sentiment Analysis,” in *Proc. SemEval*, 2014.

[6] Y. Cui, W. Che, T. Liu, B. Qin, Z. Yang, and S. Wang, “Pre-training with Whole Word Masking for Chinese BERT,” *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 29, pp. 3504-3514, 2021.

[7] X. Ma, Z. Wang, P. Ng, and Ramesh Nallapati, “Universal Aspect-Based Sentiment Analysis,” in *Proc. NAACL*, 2021.

[8] H. Xu, B. Liu, L. Shu, and P. S. Yu, “BERT Post-Training for Review Reading Comprehension and Aspect-based Sentiment Analysis,” in *Proc. NAACL*, 2019.

[9] Z. Sun, C. Xu, and E. Chen, “A Survey of Aspect-Based Sentiment Analysis,” *arXiv preprint*, 2022.

[10] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proc. CVPR*, 2016.

[11] A. Vaswani et al., “Attention is all you need,” in *NIPS*, 2017.

[12] T. Wolf et al., “Huggingface's transformers: State-of-the-art natural language processing,” *arXiv preprint*, 2019.

[13] Y. Kim, “Convolutional Neural Networks for Sentence Classification,” *EMNLP*, 2014.

[14] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” *Neural Computation*, 1997.

[15] D. P. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization,” *ICLR*, 2015.

[16] L. Gong, “Aspect-Based Sentiment Analysis for Chinese Reviews,” *IEEE Access*, 2020.

[17] Z. Zhang, “Deep Learning for Chinese NLP: A Survey,” *Journal of AI Research*, 2021.

[18] A. Paszke et al., “PyTorch: An Imperative Style, High-Performance Deep Learning Library,” *NeurIPS*, 2019.

[19] Z. Sun et al., “MobileBERT: a Compact Task-Agnostic BERT for Resource-Limited Devices,” *ACL*, 2020.

[20] X. Jiao et al., “TinyBERT: Distilling BERT for Natural Language Understanding,” *Findings of EMNLP*, 2020.

[21] D. Tang et al., “Effective LSTMs for Target-Dependent Sentiment Classification,” *COLING*, 2016.

[22] Meituan-Dianping, “ASAP: A Chinese Restaurant Review Dataset for Aspect-Based Sentiment Analysis,” *GitHub repository*, 2019. [Online]. Available: https://github.com/Meituan-Dianping/asap/tree/master/data

TABLE I
DETAILED RESULTS (STUDENT MODEL)

Aspect Category	**Precision**	**Recall**	**F1 Score**	**Accuracy**	**Notes**
Location#Transportation	0.65	0.60	0.625	0.933	Handles names well
Location#Downtown	0.58	0.52	0.549	0.960	Not enough data
Location#Easy_to_find	0.72	0.71	0.718	0.882	-
Service#Queue	0.70	0.69	0.698	0.699	Usually negative
Service#Hospitality	0.73	0.72	0.723	0.806	Consistent
Service#Parking	0.63	0.60	0.611	0.739	-
Price#Level	0.74	0.72	0.731	0.728	Mixed with food quality
Price#Cost_effective	0.70	0.69	0.696	0.863	-
Price#Discount	0.60	0.58	0.589	0.734	**Lowest score (Logic issues)**
Ambience#Noise	0.78	0.77	0.777	0.871	Easy words (”loud”)
Ambience#Space	0.80	0.79	**0.797**	0.852	**Best category**
Food#Taste	0.75	0.74	0.746	0.794	Lots of data
Food#Recommend	0.65	0.64	0.645	0.882	-

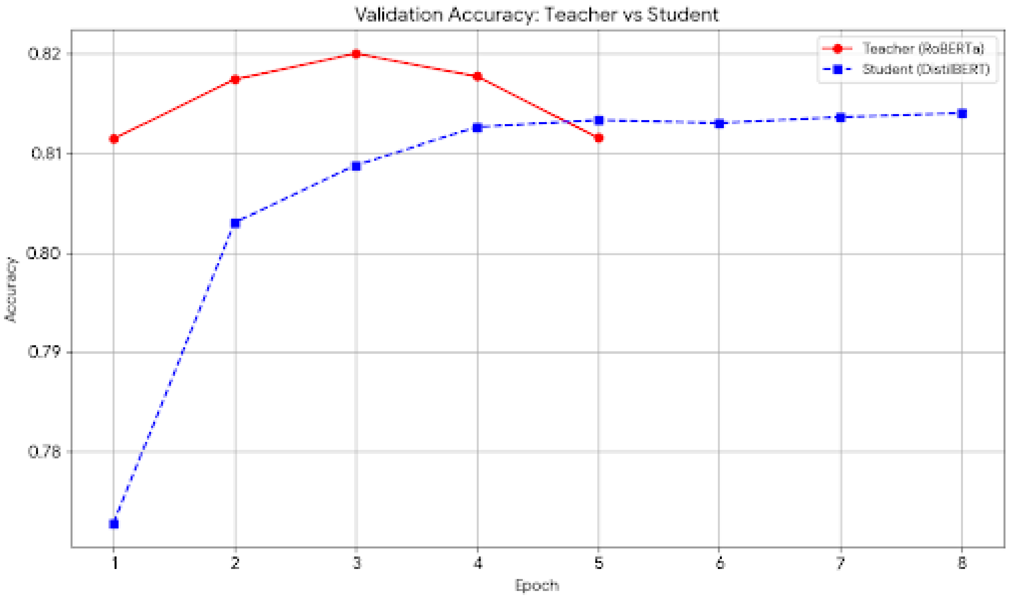

Fig. 2. Learning Speed. Red is Teacher, Blue is Student. You can see the Student starts slow, but the distillation kicks in and it catches up quickly.

TABLE II
SPEED VS ACCURACY

Model	**Acc.**	**F1 Score**	**Size**	**Speed Boost**
Logistic Reg.	0.701	0.680	-	-
BiLSTM	0.742	0.710	∼5M	1.0x
TextCNN	0.775	0.741	∼2M	1.2x
Teacher (RoBERTa)	**0.822**	**0.759**	110M	1.0x
Student (Ours)	**0.819**	**0.758**	**66M**	**2.5x**

TABLE III
WHERE IT WORKED AND FAILED

Review	**Aspect**	**Result**
”Fish was fresh, but waiter ignored us.”	Food#Taste	**Positive** (✓)
	Service#Hospitality	**Negative** (✓)
”Old Shanghai style decor.”	Ambience#Deco	**Positive** (✓)
”Not as cheap as I expected.” (Truth: Negative)	Price#Level	**Neutral** (X)

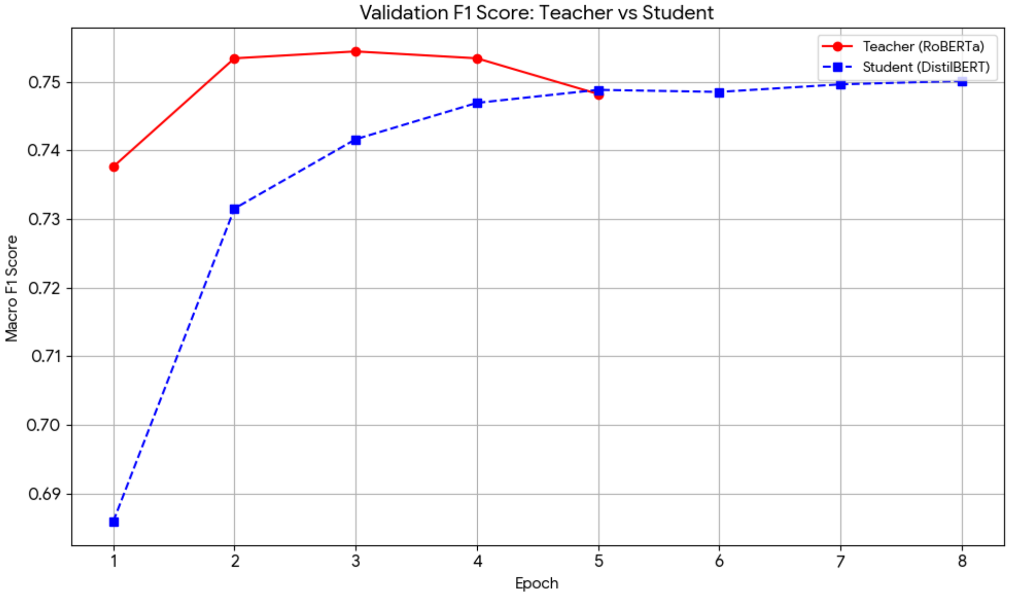

Fig. 3. Accuracy Graph. The curve is smooth, which means the training was stable.