STFXNLP at SemEval-2025 Task 11 Track A: Neural Network, Schema, and Next Word Prediction-based Approaches to Perceived Emotion Detection

Noah Murrant and Samantha Brooks and Milton King
St. Francis Xavier University
Antigonish, NS, Canada
{x2021gth,x2020ccm,mking}@stfx.ca

1 Abstract

In this work, we discuss our models that were applied to the SemEval-2025 Task 11: Bridging the Gap in Text-Based Emotion Detection (Muhammad et al., 2025b). We focused on the English dataset of track A, which involves determining what emotions the reader of a snippet of text is feeling. We applied three different types of models that vary in their approaches and reported our findings on the task's test set. We found that the performance of our models differed from each other, but neither of our models outperformed the task's baseline model.

2 Introduction

Detecting emotion in text is a complex task, as it can be challenging to identify an emotion someone is trying to convey in one snippet. It is essential to accurately identify emotions in a text as it has implications in healthcare, social science, humanities, narrative analysis, and several other fields (Muhammad et al., 2025a). It can be easier to detect emotion when someone explicitly states that they are mad or upset or uses adjectives such as delighted, pleased, or glad. "It broke my heart and nearly ruined me" clearly displays sadness and fear. However, a snippet such as "Colorado, middle of nowhere" can make it difficult to determine how people will perceive the reader's emotions. The task we focused on was Track A: Multi-label Emotion Detection, which involves determining the perceived emotions of a speaker of a snippet of text. For the English dataset, there are five possible emotions a snippet can classify as (anger, fear, joy, sadness, and surprise). This is an any-of classification; therefore, each snippet can be labelled with more than 1 emotion. In our work, we applied three different models to Task A on the dataset containing only English text. We wanted to compare three different models that varied in their approach, which included a standard feed-forward neural network-based model, a model that leveraged the idea of schemas — a key part of how humans detect, classify, and process emotions (Leahy, 2018) — and a model that relied on the next-word prediction of a large language model. Furthermore, each model relied on different types or amounts of data which is discussed in Section 4 and Section 5.

Our models did not perform well, but our best Model_FFNN model scored 82 out of 98 participants for the Task A English dataset. The Model_FFNN performed best with respect to the F1-score on the test set, achieving 64.7% [1]. It performed best on the fear class, with an F1-score of 79.03%. The fear class was the most frequent class in the training set, with 1,611 of 2,768 samples having the fear label. Our Model_FFNN's lowest F1-score was on anger, with only 49.68%. Anger had the lowest number of samples, with only 333 samples of 2768 having the anger label.

3 Background

There were three tracks for this shared task. We focused on Track A: Multi-label Emotion Detection. Within this track, we considered the English dataset. The training set contained 2,768 samples of text snippets taken from Reddit[2], as well as personal narratives, talks, and speeches. Each snippet is made up of 1 to 4 sentences. Snippet length varies between 3 words to 81 words (Muhammad et al., 2025a). The number of samples labelled with each emotion is (Anger: 333, Fear: 1611, Joy: 674, Sadness: 878, Surprise: 839). The number of samples labelled with each emotion for both the training set and test set can be seen in Table 1. Along with the provided dataset, we used WordNet synsets (Miller, 1994). We used the synsets for

[1] There is a discrepancy of approximately 0.1% between our results and those calculated by the Task 11 organizers, which we believe is due to rounding.

[2] https://www.reddit.com

Proceedings of the The 19th International Workshop on Semantic Evaluation (SemEval-2025), pages 651–656

the words anger, fear, joy, sadness, and surprise. Two of our models relied on the ability to generate embeddings. To generate these embeddings, We used BERT-Emotions-Classifier [3]. We adapted code from the Hugging face BERT documentation to create the embeddings [4]. The BERT-Emotions-Classifier was trained for the Semeval 2018 task 1 (Mohammad et al., 2018). It was trained on Tweets [5] with labels: anger, anticipation, disgust, fear, joy, love, optimism, pessimism, sadness, surprise, and trust. BERT-Emotions-Classifier can act as a full pipeline, but we only use the embeddings for our models. We did not perform additional preprocessing steps beyond what is described for each individual model, which are discussed in Section 4. Since BERT focuses on masked language modeling and is not designed for next word prediction, we used gpt-2 (Radford et al., 2019) for our model that uses next word prediction to classify text with their perceived emotions. The next word prediction setup could be viewed as completing a cloze-style (fill-in-the-blank) question, where the blank to be completed is the next word. Schick and Schütze (2021) used cloze-style questions to assist with annotating data that will then be used for training. One of the tasks that they evaluate their model on involves predicting the rating of a review. In one of our models, we compare each snippet to a schema structure. Schema theory is a cognitive psychology theory that outlines a set of learned frameworks that allow us to more quickly understand the world around us, make more accurate predictions, and even influence how we view ourselves. We can extract more information about the observation by comparing things we observe to our schemas. They help us organize and interpret information as a collection of related knowledge about a concept or entity, allowing us to quickly understand and process new information based on our past experiences (Leahy, 2018). In the case of emotion detection, by comparing a snippet to an emotion schema and gauging how similar they are, we attempt to extrapolate whether that snippet exhibits that emotion or not.

[3] https://huggingface.co/ayoubkirouane/BERT-Emotions-Classifier

[4] https://huggingface.co/docs/transformers/en/model_doc/bert

[5] https://x.com

	Anger	Fear	Joy	Sad	Surp
Training	333	1611	674	878	839
Test	322	1544	670	881	799

Table 1: Number of samples labelled with each emotion in training and test sets. Sad is the sadness class. Surp is the surprise class.

4 System Overview

This section discusses the three different models we applied to the SemEval 2025 Task 11—Track A English dataset.

4.1 Model_FFNN

This model uses five separate feed-forward neural network (FFNN) classifiers. First, the snippets are passed into the BERT-Emotions-Classifier, where the embeddings are generated. The embeddings are then passed into the five separate data loaders, one for each emotion, with their associated label, 1 or 0, that the feed-forward neural networks are trained on. Each feed-forward neural network has an input size of 768, a hidden layer size of 512, a dropout of 0.3, and an output size of one. Each FFNN comprises three fully connected layers, where the first two use a rectified linear unit and dropout, and the final layer uses a sigmoid function to produce a final probability of the associated emotion. The FFNNs do not produce a discrete label for the snippet. They only produce a probability for their corresponding emotion. The probability from each FFNN is put into a single array, where each probability represents the likelihood of anger, fear, joy, sadness, and surprise. There are five separate polynomial SVMs, one for each emotion. Each SVM takes in the array of emotion probabilities and the label corresponding to the emotion the SMV is designated. The idea behind using the SVM is to determine the proper threshold for probability and capture any relation between the probabilities. Due to the nature of the dataset, all emotions are weighted toward the false label, creating a disparity in the data and biasing the FFNNs towards classifying every snippet as having no emotion. A threshold of 50% was used for determining the final classification, but we found that the best threshold was inconsistent across all five emotions, and we began using an SVM. The SVMs capture the best threshold for each emotion's probability. For example, if the threshold for anger is

0.3, the SVM can capture that threshold. By giving the SVM all five probabilities, the SVM can detect relations between the probabilities, where if anger has a high probability, the SVM can more accurately predict whether the snippet should be classified as having the joy label. The final output from each SVM is a 1 or 0, with 1 indicating that the snippet expresses the corresponding emotion and 0 indicating that the snippet does not express the corresponding emotion.

4.2 Model_Cosine

This model uses schemas, a psychological structure used for identifying patterns and extrapolating information. For people, schemas are developed over time and encompass a large variety of information, from facial expressions, tone of voice, keywords, familiarity, and other parameters. In our research, we developed a schema by concatenating all snippets with the same label. Additionally, for every member in each WordNet (Miller, 1994) synset of each emotion, we collected the definition of each member. Senses represent the different meanings a word can exhibit and are captured by synsets. Adding the definitions of the members of the synsets of emotions provides more information to the schema. After snippets of text are concatenated and synset definitions are added, the schemas are passed into the BERT-Emotions-Classifier, and a 768-dimension embedded schema is created for each emotion. Given a snippet text to classify, we make an embedding using the BERT-Emotions-Classifier and then compare it to the five embedded schemas by measuring the cosine similarity. The cosine similarity from each comparison is put into an array and then passed to the five polynomial SVMs along with the corresponding emotions label. This is done for similar reasons it is done for the Model_FFNN. The SVMs can determine the best threshold for accurately classifying the snippet and capturing any relation between the similarities. The final output from each SVM is a 1 or 0, with 1 indicating that the snippet expresses the corresponding emotion and 0 indicating that the snippet does not express the corresponding emotion.

4.3 Model_NWP

This model uses next-word prediction to assist with predicting the perceived emotion in a snippet of text. Given a snippet of text, we strip punctuation at the beginning and end of the text and concatenate the resulting text with the string *"and I was"*. For example, if the original text was "*My stomach was hurting.*", it would become *"My stomach was hurting and I was"*. This concatenated text is then used as input to a large language model, which is used to calculate the probabilities for candidate words to be the next word[6]. We used the gpt-2(Radford et al., 2019) model from Hugging Face[7] for our language model[8]. The candidate next words are then ranked based on their probabilities from highest to lowest. We then check the ranking of words from a predefined list of words associated with the considered emotions for the task. If a word in the list is ranked higher than or equal to some threshold k, then the model predicts the emotion that corresponds with that word. The list of words and their corresponding emotions include (*angry:anger*, *afraid:fear*, *happy:joy*, *sad:sadness*, *surprised:surprise*).

5 Experimental setup

For initial training to gauge the performance of *Model_FFNN* and *Cosine*, these two models were trained on 80% (2214 samples) of the training set, then performance was calculated on the other 20% (554 samples) of the training. PyTorch[9] version 2.5.0+cpu was used for Model_FFNN. SciKit[10] Learn version 1.4.0 was used for the SVM involved with models *Model_FFNN* and *Cosine*. For training the feed-forward neural networks, data was batched into groups of 16 and shuffled. We tuned the dropout rate to 0.3 as it yielded the best F1-score and accuracy results. We found that increasing the size of the hidden layers from 256 to 512 also increased the model's performance. Initially, we used a single feed-forward neural network with an input size of 768 and an output size of 5. However, this performance was low, so we moved to five separate feed-forward neural networks binary probabilistic classifiers. We found we had the best results when using all five outputs provided by the FFNNs to determine the binary classification for each emotion. Both models *Model_FFNN* and *Cosine* were trained on the full training dataset before being submitted on the final test set.

[6] We used the implementation suggested by Ruan at https://stackoverflow.com/questions/76397904/generate-the-probabilities-of-all-the-next-possible-word-for-a-given-text to perform next word prediction.

[7] https://huggingface.co/docs/transformers/en/model_doc/gpt2

[8] We used the AutoModelForCasualLM library.

[9] https://pytorch.org/

[10] https://scikit-learn.org/stable/

Model_NWP uses a threshold k, which determines whether or not an emotion is assigned based on if their corresponding word, such as *afraid* for the fear class, appears in the top k candidate words for next word prediction. To tune our threshold *k*, we observe the performance (F1-score, precision, and recall) of the model on a set of samples that includes samples from the provided training and development set by the task prior to testing with respect to the fear class. Limiting the model to tune *k* to one emotion removes the need to have seen other emotions that might be in the test set and allows it to handle unseen emotions. However, we did observe the performance of the model with different *k* values averaged across all emotions before submitting to the task's test set to check that the model was performing near its potential across all emotions in the set that combines the provided training set and development set. We found that the best *k* value for the fear class was 110, which is true for both the set that combined the provided training and development set and only the training set[11]. 110 was also the best *k* value for the performance averaged across all emotions on both sets of samples. Further analysis showed that the best *k* value differs among different emotions. For this model, we only use a *k* value of 110 when applied to the test set.

6 Results

In this section, we discuss the performance of our models. Table 2 shows the per-emotion performance of our models. Table 3 shows the performance of our models compared against other models applied to the task with respect to the F1-score across the full test set.

Model_FFNN was our best-performing model with respect to the macro F1-score and the per-class performance. The Model_Cosine performed better than Model_FFNN in accuracy on anger and joy and only marginally worse in accuracy on the other emotions. Our models performed worse than the task's baseline model RemBERT (Chung et al., 2021) by 6.22%. The Model_Cosine and Model_FFNN performed poorly on recall except when classifying fear, with Model_Cosine R and Model_FFNN achieving a recall of 77.78% and 81.28&, respectively. Precision and recall for both models were approximately even for fear,

[11]There were some k values near 110 that had the same rounded F1-score.

	Anger	Fear	Joy	Sad	Surp
M_FFNN					
Acc	88.58	75.93	84.10	79.93	78.14
F1	49.68	79.03	65.79	66.14	62.45
R	48.45	81.28	63.13	62.20	62.95
P	50.98	79.03	68.67	70.62	61.95
M_Cosine					
Acc	90.50	74.30	84.86	79.40	73.40
F1	40.36	77.16	60.73	59.8	25.05
R	27.64	77.78	48.36	48.13	15.39
P	74.79	76.55	81.61	78.96	67.21
M_NWP					
Acc	77.05	60.03	39.61	63.57	39.61
F1	28.57	71.81	43.26	43.75	45.73
R	39.44	91.26	95.07	44.49	88.11
P	22.40	59.20	28.00	43.03	30.88

Table 2: Accuracy (ACC), F1-score (F1), Recall (R), and Precision (P) per emotion for each of our models (*Model* in the names has been shortened to M). Sad is the sadness class. Surp is the surprise class.

where fear had the highest number of labelled samples, with 1,611 of 2,768 samples having the fear label. Model_Cosine and Model_FFNN had the worst performance for recall when classifying anger, where achieved a recall of 27.64% and 48.45%, respectively. This reflects the difference in the number of samples between anger and fear. Where predicting each emotion can be viewed as a binary classification, the total number of samples trained for each binary classifier is 2,768. For each emotion, the vast majority of these samples will be considered as the negative class (labelled as not having the emotion). Our model might have been susceptible to this class imbalance in the training set. Fear was the closest to having an even distribution, with 1,611 samples labelled with fear and 1,157 as not fear. Our models achieved their best F1-score with regard to the fear class, which could be due to a more even class distribution or the increase in labelled samples. All of our models performed best when classifying fear. Model_NWP was our worse performing model with respect to the averaged F1-score and per emotion F1-score, except for the surprise class, where it outperformed Model_cosine.

Table 4 shows the co-occurrence between all emotions, meaning if we see one label, there is some probability that we are going to see another label. For example, if we see the anger label, there

Model	F1 Score (%)
Track A Best	**82.30**
Track A Baseline	70.83
Track A Average	70.58
Model_FFNN	64.70
Model_Cosine	52.62
Model_NWP	46.55

Table 3: Macro F1-scores of our models compared to other models applied to the task. There was a discrepancy between our calculated F1-score and the task organizer's F1-score of approximately 0.1%. We report the organizer's F1-score here for all models except Model_Cosine and Model_NWP, since we did not have access to those two models' performances.

	Anger	Fear	Joy	Sad	Surp
Anger	1	0.72	0.02	0.46	0.33
Fear	0.15	1	0.06	0.42	0.36
Joy	0.01	0.15	1	0.07	0.23
Sad	0.18	0.78	0.05	1	0.23
Surp	0.13	0.69	0.19	0.24	1

Table 4: Co-occurrences between each emotion, which is calculate as the P(column class | row class). Sad is the sadness class. Surp is the surprise class.

is a 72% chance we will also see the fear label, but only a 2% chance we will see the joy label. Co-occurrence is calculated by counting the number of times a label appears alongside another label, divided by the count of that label. When looking at anger, which appears 333 times, fear appears 239 times; we divide 239/333, so the co-occurrence between anger and fear is 0.72. Fear appears substantially more than anger, with 1611 samples and its co-occurrence with anger is much lower, only 0.15. Given anger, there is a high chance of seeing fear. Given fear, there is a lower chance of seeing anger. Co-occurrence did not seem to influence the models. One of the reasons for using the SVMs was to attempt to capture the relation in the co-occurrence of the emotions. Anger has a high co-occurrence with fear, being 0.72, but our models performed better on the fear class than in the anger class. The anger class also has the lowest number of positive samples. This may suggest that a higher number of positive samples (samples labelled with the target class of each binary classifier) benefits the model.

7 Conclusion

Our models underperformed compared to the task's baseline model, with our best model placing 82 out of 98 on Track A: Multi-label Emotion Detection. Model_FFNN could have underperformed due to the small number of samples. There were only 333 samples labelled as anger out of a total of 2,768 samples, which could have biased this model towards classifying samples as not anger. There is a similar case for the other emotions. Model_Cosine could have underperformed for a similar reason. Our models recruited the use of three different approaches, which included feed-forward neural networks, schemas, and next-word prediction. Although their performance was relatively poor, it would be interesting to examine the effects of data augmentation to decrease the class imbalances with the binary classification setup in future work. Additional datasets could also be recruited to provide more training data, which could assist with the supervised models Model_FFNN and Model_Cosine. Other embedding models could also be considered for our models. Future experiments could also observe if datasets containing audio can be used to assist with a similar classification task by leveraging tonality and emphases to improve the performance of the schema model.

8 Ethical Consideration

Incorrectly classifying a snippet of text with the wrong perceived emotion could have someone act or react with incorrect information. For example, if a snippet of text was incorrectly labelled with anger, then a person interpreting that could react based on incorrect knowledge, where they would otherwise react in a different manner.

References

Hyung Won Chung, Thibault Févry, Henry Tsai, Melvin Johnson, and Sebastian Ruder. 2021. Rethinking embedding coupling in pre-trained language models. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Robert L. Leahy. 2018. Introduction: Emotional schemas and emotional schema therapy. *International Journal of Cognitive Therapy*, 12(1):1–4.

George A. Miller. 1994. WordNet: A lexical database for English. In *Human Language Technology: Proceedings of a Workshop held at Plainsboro, New Jersey, March 8-11, 1994*.

Saif Mohammad, Felipe Bravo-Marquez, Mohammad Salameh, and Svetlana Kiritchenko. 2018. SemEval-2018 task 1: Affect in tweets. In *Proceedings of the 12th International Workshop on Semantic Evaluation*, pages 1–17, New Orleans, Louisiana. Association for Computational Linguistics.

Shamsuddeen Hassan Muhammad, Nedjma Ousidhoum, Idris Abdulmumin, Jan Philip Wahle, Terry Ruas, Meriem Beloucif, Christine de Kock, Nirmal Surange, Daniela Teodorescu, Ibrahim Said Ahmad, David Ifeoluwa Adelani, Alham Fikri Aji, Felermino D. M. A. Ali, Ilseyar Alimova, Vladimir Araujo, Nikolay Babakov, Naomi Baes, Ana-Maria Bucur, Andiswa Bukula, Guanqun Cao, Rodrigo Tufino Cardenas, Rendi Chevi, Chiamaka Ijeoma Chukwuneke, Alexandra Ciobotaru, Daryna Dementieva, Murja Sani Gadanya, Robert Geislinger, Bela Gipp, Oumaima Hourrane, Oana Ignat, Falalu Ibrahim Lawan, Rooweither Mabuya, Rahmad Mahendra, Vukosi Marivate, Andrew Piper, Alexander Panchenko, Charles Henrique Porto Ferreira, Vitaly Protasov, Samuel Rutunda, Manish Shrivastava, Aura Cristina Udrea, Lilian Diana Awuor Wanzare, Sophie Wu, Florian Valentin Wunderlich, Hanif Muhammad Zhafran, Tianhui Zhang, Yi Zhou, and Saif M. Mohammad. 2025a. Brighter: Bridging the gap in human-annotated textual emotion recognition datasets for 28 languages. *Preprint*, arXiv:2502.11926.

Shamsuddeen Hassan Muhammad, Nedjma Ousidhoum, Idris Abdulmumin, Seid Muhie Yimam, Jan Philip Wahle, Terry Ruas, Meriem Beloucif, Christine De Kock, Tadesse Destaw Belay, Ibrahim Said Ahmad, Nirmal Surange, Daniela Teodorescu, David Ifeoluwa Adelani, Alham Fikri Aji, Felermino Ali, Vladimir Araujo, Abinew Ali Ayele, Oana Ignat, Alexander Panchenko, Yi Zhou, and Saif M. Mohammad. 2025b. SemEval task 11: Bridging the gap in text-based emotion detection. In *Proceedings of the 19th International Workshop on Semantic Evaluation (SemEval-2025)*, Vienna, Austria. Association for Computational Linguistics.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Timo Schick and Hinrich Schütze. 2021. Exploiting cloze-questions for few-shot text classification and natural language inference. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 255–269, Online. Association for Computational Linguistics.