AI Love Coach

Clervilsson Christelle

Research Project

September 4, 2025

1 Introduction

In the field of Natural Language Processing (NLP), many research topics are important and fascinating. One of the main challenges is the natural complexity of language: the meaning of words can change depending on context. Sometimes it is even difficult for us humans to think human.

For this project, we decided to explore the intersection between NLP and couple psychology, an area that combines technical challenges with human relevance in the real-word. The motivation comes from the idea of applying NLP methods to love coaching: analyzing conversations between partners to detect emotions, identify relational problems, and ultimately provide supportive guidance.

The primary goal of this work is therefore to design an artificial intelligence system capable of analyzing couple dialogues and generating meaningful relational insights. This involves multiple steps, from data preprocessing and problem detection to emotion recognition and response generation using large language models.

2 Steps

1. Dataset Selection and Preparation
2. Problem Label Detection
3. Emotion recognition

2.1 Dataset Selection and Preparation

Three datasets were used for this project, each serving a specific purpose.

The first dataset is **DailyDialog**. This dataset is a collection of multi-turn dialogues that aim to accurately represent natural human conversations. It includes human-written conversations, which make the language more natural and realistic. Each dialogue contains two or more participants. In addition, each conversation has been manually labeled with information on intention and communication emotion, providing valuable information about the conversations.

The dataset contains three columns:

- **dialog**: Contains the actual conversation between participants, in text format.
- **act**: Represents communication intention labels for each utterance in the dialogue, categorizing each utterance based on its intention.
- **emotion**: Contains emotion labels for each utterance, representing the emotions expressed.

DailyDialog is divided into three parts: training, validation, and test sets. We chose this dataset to understand the structure of dialogues and to explore the detection of couple-related problems. However, we observed that this dataset is not specifically focused on couple discussions, and the topics of the dialogues are often unrelated to the problem labels. Consequently, using unsupervised learning to annotate the entire dataset would be challenging.

Couple relationship datasets are difficult to obtain due to privacy and intimacy concerns. Therefore, we created a small custom dataset inspired by Reddit discussions about couple problems. This dataset contains four columns: **speaker**, **text**, **emotion**, and **problem**. It is intended to train our problem detection model on domain-specific data.

The last dataset is **GoEmotions**. GoEmotions is a corpus of 58,000 carefully curated comments extracted from Reddit, annotated with 27 emotion categories or neutral.

- Number of examples: 58,009
- Number of labels: 27 + Neutral
- Maximum sequence length in training and evaluation datasets: 30

The dataset contains a train/test/validation split:

- Training set: 43,410 examples
- Test set: 5,427 examples
- Validation set: 5,426 examples

The original 27 emotion categories are: admiration, amusement, anger, annoyance, approval, caring, confusion, curiosity, desire, disappointment, disapproval, disgust, embarrassment, excitement, fear, gratitude, grief, joy, love, nervousness, optimism, pride, realization, relief, remorse, sadness, and surprise. For this project, we merged these 27 emotions into 7 categories: **joy, anger, sadness, love, fear, surprise and neutral**.

2.1.1 Data Pre-processing

We first cleaned the DailyDialog dataset by removing extra spaces, normalizing the text spacing, and eliminating duplicates to ensure a clean dataset. We also removed emojis and URLs.

Additional preprocessing steps were applied later, which were crucial for subsequent tasks. These included removing stop words and performing lemmatization. Stop words are very frequent words that do not add meaningful information to the corpus, such as "I", "am", or "no". Lemmatization transforms words to their root forms; for example, "feelings" becomes "feel". We also removed words that were either too short (less than

3 letters) or excessively long (more than 20 letters), as they were unlikely to contribute useful information.

Since not all conversations in DailyDialog were relevant, we applied a filtering process to select only the most interesting dialogues. The first step involved computing TF-IDF scores to identify the most important words in the corpus. These scores were then used to select dialogues that could be related to couple discussions.

2.2 Problem Detection

2.2.1 TF-IDF and K-Means

TF-IDF (Term Frequency–Inverse Document Frequency) is a numerical statistic that reflects the importance of a word in a document relative to the corpus. Term Frequency (TF) increases with the frequency of a word in a document, while Inverse Document Frequency (IDF) assigns higher scores to words that are rare across the corpus, making them more distinctive.

To extract themes from the corpus, we applied K-Means clustering. The main challenge was determining the optimal number of clusters. Initially, we used the elbow method, which plots the explained variance as a function of the number of clusters. However, the results for 2 to 200 clusters were inconclusive, as shown in Figure 1.

To complement the elbow method, we also evaluated the clustering using the silhouette score, which measures how similar an element is to its own cluster compared to other clusters. High silhouette values indicate a well-clustered configuration. Unfortunately, the silhouette analysis revealed that the clustering configuration was not optimal for this dataset, as illustrated in Figure 2.

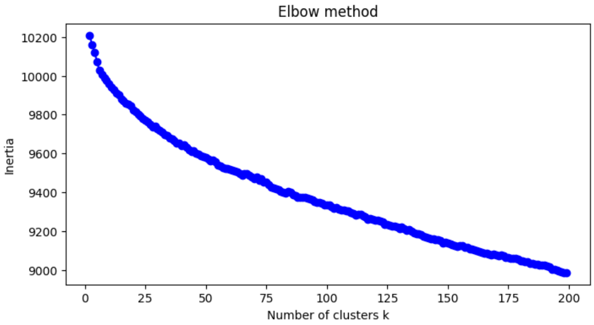

Figure 1: Elbow method for K-Means clustering

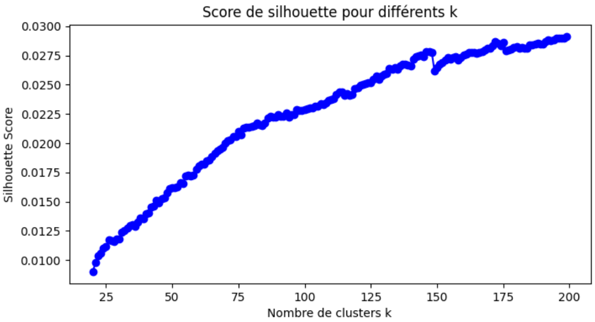

Figure 2: Silhouette score for different values of K

2.2.2 Latent Dirichlet Allocation (LDA)

LDA is a probabilistic generative model used in topic modeling to discover hidden topics in a corpus. It assumes that each document is a mixture of multiple topics, and each topic is represented as a distribution of words. After training, LDA provides insight into the topics that dominate each document.

We applied LDA to the DailyDialog dataset to explore whether relationship-related topics could be identified.

When the stop words were not removed, the resulting topics were:

Topic 0: look, dont, just, oh, think, good, want, im, yes, like **Topic 1:** help, like, time, thank, mr, good, im, need, right, yes **Topic 2:** time, good, work, really, did, just, think, dont, know, im **Topic 3:** time, yes, just, good, going, people, really, know, think, like **Topic 4:** room, help, good, ok, want, need, thank, sir, yes, like

After the stop words were removed, the results improved slightly:

Topic 0: yeah, day, good, love, look, oh, time, like, know, think **Topic 1:** want, look, car, house, yes, people, know, think, play, like **Topic 2:** help, good, use, account, money, yes, know, like, want, need **Topic 3:** new, day, company, thank, like, good, time, yes, job, work **Topic 4:** leave, let, know, oh, time, look, thank, yes, right, bus **Topic 5:** let, ill, thank, think, buy, good, yes, look, want, like **Topic 6:** yes, school, movie, food, chinese, think, know, english, good, like **Topic 7:** country, year, know, product, people, business, price, good, think, company **Topic 8:** pay, help, right, like, check, ok, thank, yes, sir, room **Topic 9:** oh, tomorrow, right, time, ill, come, sorry, mr, yes, thank

Although these results contained more meaningful words, none of the topics were directly related to relationship issues. For this reason, we abandoned the DailyDialog dataset for problem detection and decided to create our own small dataset inspired by online discussions of couple issues.

—

2.2.3 Model Evaluation with TF-IDF Features

All of these tests and observations were first conducted on DailyDialog in an attempt to extract relationship-related themes. However, when this approach became too complex and inconclusive, we turned to our own dataset and trained multiple models on it. Using TF-IDF features, we observed that Random Forest and Gradient Boosting provided the best performance, as shown in Table 1.

Table 1: Comparison of TF-IDF based models for couple issue classification

Model	Accuracy	Macro F1	Weighted F1
Logistic Regression	0.82	0.80	0.79
SVM	0.55	0.53	0.45
Random Forest	**0.82**	**0.87**	**0.82**
Gradient Boosting	**0.82**	**0.87**	**0.82**

Despite these results, TF-IDF has clear limitations: it only considers individual word frequencies without capturing semantics or contextual meaning. Since relationship discussions often involve nuance and ambiguity, we explored more advanced representations.

2.2.4 Contextual Embeddings and Classification Techniques

We next moved from TF-IDF to word embeddings and, ultimately, to contextual embeddings such as BERT. Unlike TF-IDF, contextual embeddings capture both semantic similarity between words and the context in which they appear, which is crucial to understanding human dialogue.

2.2.5 BERT Embeddings

Bidirectional Encoder Representations from Transformers (BERT) is a pre-trained transformer that is trained on the task of masked language modeling. When BERT is provided with a sentence as input, it outputs a contextual vector representation of every token in the input sequence. It is also a convention to use the special [**CLS**] token embedding as a sentence summary representation.

However, BERT embeddings were specifically not built to understand semantic meaning at the sentence level. The [**CLS**] representation will mix syntactic and context information, but does not explicitly represent sentence-level similarity. BERT embeddings are very effective when fine-tuned for prediction tasks such as emotion detection, sentiment analysis, or relational problem detection.

2.2.6 Sentence Embeddings

Sentence embeddings aim to capture the semantic meaning of an entire sentence in a single dense vector.They are particularly practical for use in applications such as semantic textual similarity, clustering of dialogues, and information retrieval. They provide more stable sentence meaning representations compared to BERT embeddings with minimal fine-tuning. This makes them a perfect choice when working with tasks that consider comparing or clustering sentences.

2.2.7 Classical Machine Learning Models on Top of Embeddings

After the embeddings are obtained , they can be used as input for regular machine learning classifiers. This is a light and fast method for small or medium-sized datasets. The models to be considered in this project are:

- **Logistic Regression (LogReg)**: A light linear model for linearly separable embeddings. It is fast, interpretable, and generally a good baseline. ¡/itemize¿.

- **Support Vector Machines (SVM)**: Perform well in high-dimensional space, but computationally expensive for large datasets. SVMs generalized very poorly on sentence embeddings in our experiments.

- **Random Forest (RF)**: An ensemble algorithm that uses multiple decision trees. It performs well with non-linear datasets.

- **Gradient Boosting (GB)**: A second ensemble technique, typically more powerful than RF, but slower to train. It worked well with the TF-IDF features.

- **Multi-Layer Perceptron (MLP)**: A basic neural network capable of modeling non-linear interaction in embeddings. When given enough data, MLPs outperform traditional models with minimal overhead.

This multi-step approach, which transforms text into numeric vectors using embeddings and then using machine learning classifiers, allowed us to test out numerous relational problems and sentiment identification techniques while balancing performance and computational cost.

2.2.8 Model Evaluation with Contextual Embeddings

Table 2 summarizes the performance of different classifiers using BERT embeddings. Logistic Regression, Random Forest, Gradient Boosting, and MLP performed similarly well, whereas SVM failed to adapt to high-dimensional embeddings.

Table 2: Comparison of BERT embeddings-based classifiers for couple issue classification

Classifier	**Accuracy**	**Macro F1**	**Weighted F1**
Logistic Regression	**0.82**	**0.80**	**0.79**
SVM	0.18	0.17	0.08
Random Forest	**0.82**	**0.80**	**0.79**
Gradient Boosting	**0.82**	**0.80**	**0.79**
MLP	**0.82**	**0.80**	**0.79**

Table 3: Comparison of sentence embeddings-based classifiers for couple issue classification

Classifier	**Accuracy**	**Macro F1**	**Weighted F1**
Logistic Regression	**0.91**	**0.94**	0.91
SVM	**0.91**	**0.94**	0.91
Random Forest	0.91	0.77	**0.95**
Gradient Boosting	0.45	0.44	0.52
MLP	**0.91**	**0.94**	0.91

These results 3 confirmed that contextual embeddings are better suited for this task, as they provide the semantic understanding necessary to detect subtle relational problems.

2.3 Emotion recognition

The next step of the project was to detect the emotions present in the conversations. This step required the most time since we needed a larger dataset to train the model. For this task, we used the *GoEmotions* dataset. The preprocessing followed the same steps as for the DailyDialog dataset, but we also harmonized the labels to simplify training. As explained in the presentation section of the datasets, we reduced the 27 original emotion categories to 7, to make the evaluation more manageable. Each emotion was then represented as a 7-dimensional vector, which served as input for training.

2.3.1 Large Language Models

To detect emotions in couple dialogues, we had to select a suitable large language model (LLM) capable of achieving strong performance after fine-tuning. Multiple models were considered, but due to the limited computational resources and memory constraints of Google Colab, we had to carefully select models that were small enough to train within the

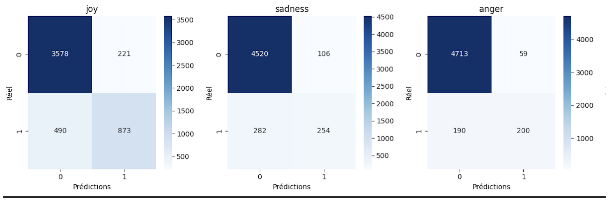

Figure 3: RoBERTa Confusion Matrix (example 1).

session limits. Some models would have required more than 13 hours of training, leading to frequent session crashes, so we focused on lightweight yet effective architectures. We fine-tuned these models using the GoEmotions dataset.

Model	**Size**	**Type**	**Strengths**	**Weaknesses**
TinyBERT	∼15M params (60 MB)	Distilled BERT	Very lightweight, fast fine-tuning, strong for sentiment/emotion tasks.	Loses nuance in longer or complex dialogues.
DistilBERT	∼66M params (250 MB)	Distilled BERT	Good balance of speed and accuracy, widely used with strong support.	Limited context length (512 tokens).
BERT Mini Sentiment	∼10M params (35 MB)	Mini BERT (pre-finetuned)	Already fine-tuned for emotion/sentiment, extremely lightweight.	Restricted to a few basic emotion categories (joy, sadness, anger, etc.).
RoBERTa-base	∼125M params (500 MB)	Transformer (BERT variant)	Stronger performance than BERT on text classification.	Slower training, heavier to fine-tune on Colab Free.
T5-Small	∼60M params (220 MB)	Text-to-Text Transformer	Flexible: handles both classification and generation (useful for "AI Love Coach").	Longer training time, slightly less efficient than distilled BERTs.

Table 4: Comparison of candidate NLP models for emotion detection and dialogue fine-tuning (Colab-friendly).

Based on these considerations, we fine-tuned the four first models to compare their performance. As expected, DistilBERT required the longest training time, but it also delivered the best overall results.

Table 5: Comparison of model performance and training time on GoEmotions (7 emotion classes).

Model	**Accuracy**	**Weighted F1**	**Training Time (hrs)**
RoBERTa-base	0.6185	0.7144	4.25
DistilBERT-base	**0.6314**	**0.719**	7.67
BERT Mini Sentiment	0.5868	0.6746	0.70
TinyBERT	0.5558	0.6383	0.23

To better analyze performance across different emotion categories, we generated confusion matrices for each model. These allow us to visualize the distribution of predictions (True Positive, False Positive, True Negative, False Negative) for each emotion. For clarity, we only present the matrices for the two best models, the RoBERTa-base and the DistilBERT-base.

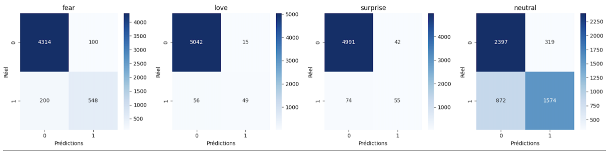

Figure 4: RoBERTa Confusion Matrix (example 2).

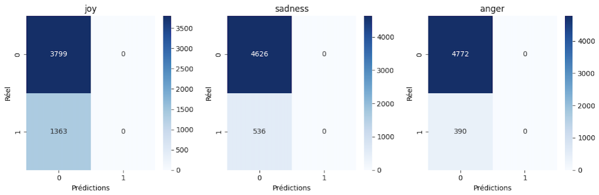

Figure 5: DistilBERT Confusion Matrix (example 1).

2.3.2 Hyper parameters tuning

We also tried to change some parameters to see if we could get better results. The first case 6 takes 3 hours to complete, while the second takes close to 7 hours.

Table 6: Hyperparameter tuning summary for DistilBERT-base-uncased with 0.001 weight decay

Epoch	Learning Rate	Batch Size (Train/Eval)	Accuracy	Weighted F1
1	1e-4	16 / 16	0.583	0.669
2	1e-4	16 / 16	0.582	0.677
3	1e-4	16 / 16	0.590	0.680

Table 7: Hyperparameter tuning summary for DistilBERT-base-uncased 0.01 weight decay

Epoch	Learning Rate	Batch Size (Train/Eval)	Accuracy	Weighted F1
1	2e-5	16 / 16	0.619	0.707
2	2e-5	16 / 16	0.631	0.719
3	2e-5	16 / 16	-	-

3 Conclusion

In this project, our goal was to create an AI tool that could be used as a ”love coach.” The general idea was to design a system capable of analyzing couple conversations, de-

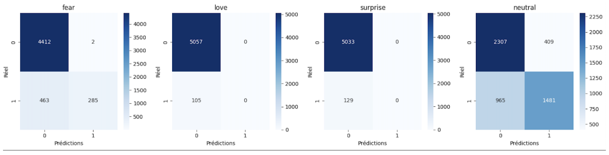

Figure 6: DistilBERT Confusion Matrix (example 2).

tecting relational problems, and recognizing the emotions expressed during interactions. However, we faced several important challenges throughout the process.

The first major difficulty was obtaining a dataset that was sufficiently related to couple problems to train our models. Although we explored several existing datasets, the results were not conclusive because most of them were too generic like DailyDialog. For this reason, we decided to build our own mini-dataset inspired by online couple discussions like Reddit. Although this allowed us to continue the project, the dataset we created was still limited in both size and diversity. This directly impacted the performance of our models, since even advanced context embedding models such as BERT require a large and diverse dataset to achieve robust results.

Another challenge comes from the complexity of human communication. Conversations, especially in the context of relationships, are often ambiguous, filled with implicit meaning, and strongly dependent on personal codes or styles of expression. This becomes even more difficult in text-based communication on social networks, where spelling variations, abbreviations, and informal expressions are frequent. For this reason, if our model were tested directly on a real-world dataset such as Instagram conversations, its ability to detect relational problems would likely be limited.

However, our experiments with emotion recognition were more promising. Using the GoEmotions dataset, we achieved reasonably good results, showing that the models were able to capture the main emotional signals expressed in text. However, we believe that with a larger dataset and greater computational capacity, the system could be significantly improved, especially in handling the subtleties and complexities of human emotions in couple dialogues.

References

[1] Yao Fu, Shaoyang Yuan, Chi Zhang, and Juan Cao, *Emotion Recognition in Conversations: A Survey Focusing on Context, Speaker Dependencies, and Fusion Methods*, arXiv preprint arXiv:2203.00000, 2022.

[2] Endang Wahyu Pamungkas, *Emotionally-Aware Chatbots: A Survey*, University of Turin, Turin, Italy, 2021.